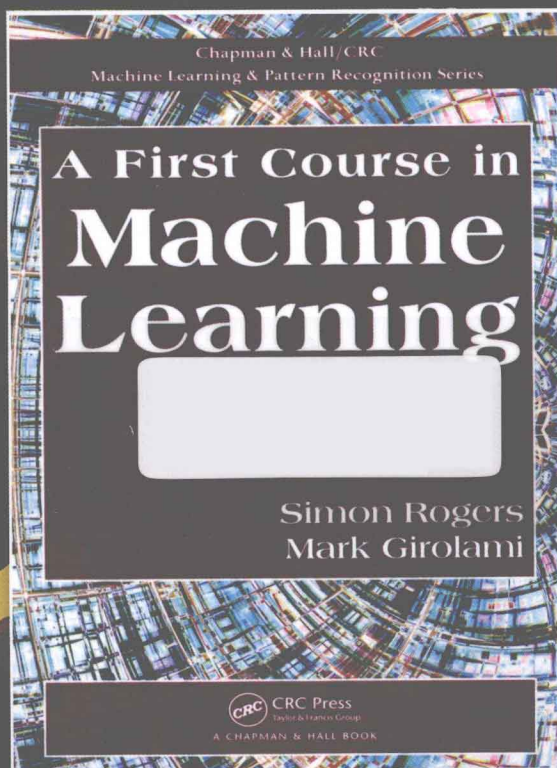


机器学习基础教程

(英) Simon Rogers Mark Girolami 著

郭茂祖 王春宇 刘扬 刘晓燕 译

A First Course in Machine Learning



机器学习基础教程

A First Course in Machine Learning

本书是一本机器学习入门教程，包含了数学和统计学的核心技术，用于帮助理解一些常用的机器学习算法。书中展示的算法涵盖了机器学习的各个重要领域：分类、聚类和投影。本书对一小部分算法进行了详细描述和推导，而不是简单地将大量算法罗列出来。

本书通过大量的MATLAB/Octave脚本将算法和概念由抽象的等式转化为解决实际问题的工具，利用它们读者可以重新绘制书中的插图，并研究如何改变模型说明和参数取值。

本书特色

- 介绍机器学习技术及应用的主要算法和思想。
- 为读者进一步探索机器学习领域中的特定方向提供起点。
- 不需要太多的数学知识，穿插在文中的注解框提供相应的数学解释。
- 每章末均包含练习。

作者简介

Simon Rogers 英国格拉斯哥大学计算机科学学院讲师，主讲硕士生的机器学习课程。Rogers博士是机器学习领域的一位活跃研究者，研究兴趣包括代谢组学数据分析和概率机器学习技术在人机交互领域的应用。



Mark Girolami 英国伦敦大学学院（UCL）统计系主任和计算机科学系荣誉教授，并担任计算统计学和机器学习研究中心主任。他还是英国统计协会研究组成员，英国工程和科学研究委员会高级研究员，英国工程技术学会会员，爱丁堡皇家学会院士。



客服热线：(010) 88378991 88361066
购书热线：(010) 68326294 88379649 68995259
投稿热线：(010) 88379604

数字阅读：www.hzmedia.com.cn
华章网站：www.hzbook.com
网上购书：www.china-pub.com

上架指导：计算机/人工智能/机器学习

ISBN 978-7-111-40702-7



9 787111 407027 >

定价：45.00元

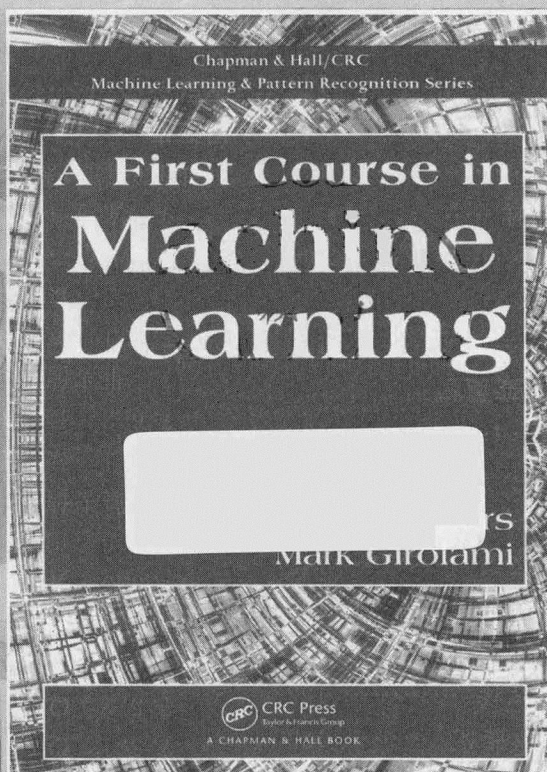
计 算 机 科 学 丛 书

机器学习基础教程

(英) Simon Rogers Mark Girolami 著

郭茂祖 王春宇 刘扬 刘晓燕 译

A First Course in Machine Learning



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

机器学习基础教程/ (英) 罗杰斯 (Rogers, S.), (英) 吉罗拉米 (Girolami, M.) 著; 郭茂祖等译. —北京: 机械工业出版社, 2013. 10

(计算机科学丛书)

书名原文: A First Course in Machine Learning

ISBN 978-7-111-40702-7

I. 机… II. ①罗… ②吉… ③郭… III. 机器学习—教材 IV. TP181

中国版本图书馆 CIP 数据核字 (2013) 第 109878 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2012-0581

A First Course in Machine Learning by Simon Rogers, Mark Girolami (ISBN: 978-1-4398-2414-6).

Copyright © 2012 by Taylor & Francis Group, LLC.

Authorized translation from the English language edition published by CRC Press, part of Taylor & Francis Group LLC. All rights reserved.

China Machine Press is authorized to publish and distribute exclusively the Chinese (Simplified Characters) language edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Copies of this book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal.

本书原版由 Taylor & Francis 出版集团旗下 CRC 出版公司出版, 并经授权翻译出版。版权所有, 侵权必究。

本书中文简体字翻译版授权由机械工业出版社独家出版并限在中国大陆地区销售。未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何内容。

本书封面贴有 Taylor & Francis 公司防伪标签, 无标签者不得销售。

本书介绍机器学习技术及应用的主要算法, 重点讲述理解主流的机器学习算法所需的核心数学和统计知识。书中介绍的算法涵盖机器学习的主要问题: 分类、聚类和投影。由于本书是机器学习基础课程的教材, 所以尽量减少了数学难度, 仅对一小部分重要算法给出详细的描述和推导, 而对大部分算法仅给出简单介绍, 目的在于使学生打好基础, 增强信心和兴趣, 鼓励他们进一步学习该领域的高级主题或从事相关研究工作。

本书是机器学习导论课程教材, 适合作为计算机、自动化及相关专业高年级本科生或研究生的教材, 也可供研究人员和工程技术人员参考。

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 盛思源

藁城市京瑞印刷有限公司印刷

2014 年 1 月第 1 版第 1 次印刷

185mm×260mm·12.5 印张

标准书号: ISBN 978-7-111-40702-7

定 价: 45.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域中取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章教育

华章科技图书出版中心

机器学习起初只是人工智能 (AI) 的一个研究分支, 随着其他研究分支的成熟发展或逐步淡化, 目前机器学习发展成为 AI 中最具活力的研究方向。一方面它源于机器学习, 已经成为人工智能理论研究与应用研究的桥梁; 另一方面, 随着计算机技术的发展, 机器学习也日益成为计算机科学的重要研究领域之一。此外, 模式识别与数据挖掘的核心算法大多也与机器学习有关。

机器学习作为人工智能理论研究的一部分, 需要一定的数学知识作为基础。本书就是为计算机等信息类专业的学生理解最流行的机器学习算法提供核心数学知识和统计技术。本书并没有面面俱到地介绍所有的机器学习算法, 而是给出部分代表性算法的核心思想及详细描述。最后, 本书主要涉及基于示例的归纳学习, 至于神经网络等进化学习以及关于 agent 与环境交互的强化学习这两大类机器学习的相关内容, 请读者参阅其他书籍。

本书共 7 章。第 1、2 章介绍如何选择线性模型参数以对观测数据做出预测。第 1 章给出通过最小化损失函数来学习模型参数的方法。第 2 章介绍最大似然函数的方法。第 3 章介绍机器学习中的贝叶斯方法。第 4 章介绍计算后验的三种近似方法。第 5 章及后续各章涉及机器学习领域分类、聚类和预测方面的主要算法, 其中第 5 章关注监督学习; 第 6、7 章介绍无监督学习, 第 6 章研究 K 均值和混合模型两种聚类方法, 第 7 章介绍通过将高维数据投影到一个低维空间, 对数据进行可视化或特征选择的方法。本书还包括词汇表和索引。

本书适合作为高等院校计算机、自动化等专业本科生及研究生的机器学习教材。同时, 本书也是机器学习领域的研究者或者那些想了解和应用当前机器学习技术的工作人员的一本宝贵的参考资料。

本书的翻译工作由郭茂祖主持, 郭茂祖审校了全部译稿, 邢林林负责校对。其中, 郭茂祖翻译了前言和第 1 章, 王春宇翻译了第 2、3 章, 刘扬翻译了第 4、5 章和词汇表、索引, 刘晓燕翻译了第 6、7 章。在本书的翻译过程中, 王娟、刘茹、徐云刚、滕志霞、李艳娟、车凯、程爽、史文丽、孟宪伟、代启国、李晋、吴伟宁、徐立秋给予了很多帮助, 对他们表示由衷的感谢。

目前机器学习日益成为计算机科学重要的实践、研究与开发领域之一，一方面这反映在它的学术研究规模上，另一方面反映在新的机器学习从业人员遍布于主要的国际银行和金融机构，以及微软、谷歌、雅虎和亚马逊等公司。

从某种角度来讲，这种发展源于人们对世界认知方式的数量和种类的增加。一个特别显著的例子是，在首个基因组测序完成之前，不断涌现出了各种生物检测新技术。不久前，检测生物体的复杂分子状态是难以想象的，因为这已经远远超出了我们的认识能力。现在，机器学习方法在生物体中分子结构提取方面的广泛应用，使其成为可能。

本书改编自英国格拉斯哥大学计算机科学学院机器学习课程的讲义，该课程包括 20 学时的授课和 10 学时的实验，面向高年级本科生开设并由研究生讲授。如此少的教学时数不可能涵盖机器学习所有的内容，所以该课的目的是为理解流行的机器学习算法提供核心数学知识和统计技术，并描述其中部分算法，这些算法涵盖了机器学习中的分类、聚类和投影等主要问题。通过本课程的学习，学生应该具备通过考察机器学习相关文献来寻求适合他们所需方法的知识 and 能力，希望本书的读者也能做到这一点。

鉴于选学该课学生的数学水平参差不齐，我们只假定需要很少的数学知识，计算机科学、工程类、物理学（或其他数值处理类学科）的本科生阅读本书应该没有问题，没有以上经历的读者也可以阅读本书，因为穿插在文中的注解框内给出了相应的数学解释。此外，突出强调了重要公式（公式加阴影），在继续阅读前，花些时间理解这些公式是值得的。

选学该课的学生通常会发现其中的实践环节非常有用，实验有助于将涉及的各种算法和概念由抽象的等式转化为解决实际问题的工具。我们已通过大量的 MATLAB[®]/Octave[⊖] 软件脚本完成以上转化，这些脚本可通过相关的网页并参考本书正文获得，利用它们读者能够重新绘制书中的插图，并研究如何改变模型说明和参数取值。

最后，本书选择的机器学习方法是我们认为学生应该掌握的，在有限的篇幅和时间内，更有必要给出一小部分算法的详细描述和研究进展，而不是泛泛地描述许多算法，因而多数读者在本书中可能找不到他们最喜欢的算法！

Simon Rogers

Mark Girolami

⊖ 免费数学软件环境，源于 www.gnu.org/software/octave/。

目 录

A First Course in Machine Learning

出版者的话

译者序

前言

第 1 章 线性建模：最小二乘法 1

1.1 线性建模 1

1.1.1 定义模型 2

1.1.2 模型假设 2

1.1.3 定义什么是好的模型 3

1.1.4 最小二乘解：一个有效的 例子 4

1.1.5 有效的例子 7

1.1.6 奥运会数据的最小二乘 拟合 8

1.1.7 小结 9

1.2 预测 9

1.2.1 第二个奥运会数据集 10

1.2.2 小结 12

1.3 向量/矩阵符号 12

1.3.1 例子 17

1.3.2 数值的例子 18

1.3.3 预测 19

1.3.4 小结 19

1.4 线性模型的非线性响应 19

1.5 泛化与过拟合 22

1.5.1 验证数据 22

1.5.2 交叉验证 23

1.5.3 K 折交叉验证的计算 缩放 25

1.6 正则化最小二乘法 25

1.7 练习 27

其他阅读材料 28

第 2 章 线性建模：最大似然方法 29

2.1 误差作为噪声 29

2.2 随机变量和概率 30

2.2.1 随机变量 30

2.2.2 概率和概率分布 31

2.2.3 概率的加法 32

2.2.4 条件概率 32

2.2.5 联合概率 33

2.2.6 边缘化 34

2.2.7 贝叶斯规则介绍 36

2.2.8 期望值 37

2.3 常见的离散分布 39

2.3.1 伯努利分布 39

2.3.2 二项分布 39

2.3.3 多项分布 40

2.4 连续型随机变量——概率密度 函数 40

2.5 常见的连续概率密度函数 42

2.5.1 均匀密度函数 42

2.5.2 β 密度函数 43

2.5.3 高斯密度函数 44

2.5.4 多元高斯 44

2.5.5 小结 46

2.6 产生式的考虑（续） 46

2.7 似然估计 47

2.7.1 数据集的似然值 48

2.7.2 最大似然 49

2.7.3 最大似然解的特点 50

2.7.4 最大似然法适用于复杂 模型 52

2.8 偏差-方差平衡问题 53

2.9 噪声对参数估计的影响 53

2.9.1 参数估计的不确定性 54

2.9.2 与实验数据比较 57

2.9.3 模型参数的变异性 ——奥运会数据 58

2.10 预测值的变异性	59	4.4 拉普拉斯近似	100
2.10.1 预测值的变异性——一个例子	59	4.4.1 拉普拉斯近似实例：近似 γ 密度	101
2.10.2 估计值的期望值	61	4.4.2 二值响应模型的拉普拉斯近似	102
2.10.3 小结	63	4.5 抽样技术	103
2.11 练习	63	4.5.1 玩飞镖游戏	104
其他阅读材料	64	4.5.2 Metropolis-Hastings 算法	105
第3章 机器学习的贝叶斯方法	66	4.5.3 抽样的艺术	110
3.1 硬币游戏	66	4.6 小结	111
3.1.1 计算正面朝上的次数	67	4.7 练习	111
3.1.2 贝叶斯方法	67	其他阅读材料	111
3.2 精确的后验	70	第5章 分类	113
3.3 三个场景	71	5.1 一般问题	113
3.3.1 没有先验知识	71	5.2 概率分类器	113
3.3.2 公平的投币	76	5.2.1 贝叶斯分类器	114
3.3.3 有偏的投币	78	5.2.2 逻辑回归	121
3.3.4 三个场景——总结	80	5.3 非概率分类器	123
3.3.5 增加更多的数据	80	5.3.1 K 近邻算法	123
3.4 边缘似然估计	80	5.3.2 支持向量机和其他核方法	125
3.5 超参数	82	5.3.3 小结	132
3.6 图模型	83	5.4 评价分类器的性能	133
3.7 奥运会 100 米数据的贝叶斯处理实例	84	5.4.1 准确率——0/1 损失	133
3.7.1 模型	84	5.4.2 敏感性和特异性	133
3.7.2 似然估计	85	5.4.3 ROC 曲线下的区域	134
3.7.3 先验概率	85	5.4.4 混淆矩阵	135
3.7.4 后验概率	85	5.5 判别式和产生式分类器	136
3.7.5 1 阶多项式	87	5.6 小结	136
3.7.6 预测	89	5.7 练习	136
3.8 边缘似然估计用于多项式模型阶的选择	90	其他阅读材料	137
3.9 小结	91	第6章 聚类分析	138
3.10 练习	91	6.1 一般问题	138
其他阅读材料	92	6.2 K 均值聚类	139
第4章 贝叶斯推理	94	6.2.1 聚类数目的选择	141
4.1 非共轭模型	94	6.2.2 K 均值的不足之处	141
4.2 二值响应	94	6.2.3 核化 K 均值	141
4.3 点估计：最大后验估计方案	96	6.2.4 小结	144

6.3 混合模型	144	7.3.2 小结	166
6.3.1 生成过程	144	7.4 变分贝叶斯	166
6.3.2 混合模型似然函数	146	7.4.1 选择 $Q(\theta)$	167
6.3.3 EM 算法	146	7.4.2 优化边界	168
6.3.4 例子	151	7.5 PCA 的概率模型	168
6.3.5 EM 寻找局部最优	153	7.5.1 $Q_z(\tau)$	169
6.3.6 组分数目的选择	153	7.5.2 $Q_{x_n}(x_n)$	170
6.3.7 混合组分的其他形式	154	7.5.3 $Q_{w_n}(w_n)$	171
6.3.8 用 EM 估计 MAP	156	7.5.4 期望值要求	171
6.3.9 贝叶斯混合模型	157	7.5.5 算法	172
6.4 小结	157	7.5.6 例子	173
6.5 练习	157	7.6 缺失值	174
其他阅读材料	158	7.6.1 缺失值作为隐变量	176
第 7 章 主成分分析与隐变量模型	159	7.6.2 预测缺失值	176
7.1 一般问题	159	7.7 非实值数据	177
7.2 主成分分析	161	7.7.1 概率 PPCA	177
7.2.1 选择 D	164	7.7.2 议会数据可视化	180
7.2.2 PCA 的局限性	165	7.8 小结	184
7.3 隐变量模型	165	7.9 练习	184
7.3.1 隐变量模型中的混合 模型	165	其他阅读材料	184
		词汇表	185
		索引	188

线性建模：最小二乘法

在有着广泛应用的机器学习中，一个重要且普遍的问题是学习或者推断属性变量与相应的响应变量或目标变量之间的函数关系，使得对任何一个属性集合，我们可以预测其响应。例如，我们可能想要建立一个能够执行疾病诊断的模型。为了构建这个模型，需要使用一个数据集，这个数据集是从已知疾病状态（响应，健康或患病）的患者中得到的测量（属性，如血压、心率、体重等）的集合。在完全不同的例子中，我们希望给顾客提出建议。在这种情况下，我们能够建立一个关于某个顾客以前买过物品的描述（属性）和该顾客最终是否喜欢该产品（响应）的模型。这个模型可以帮助我们预测顾客可能喜欢的物品，并因此进行推荐。这一章将涉及许多更重要的应用领域。

1.1 线性建模

首先，通过一个实际例子来考虑机器学习最直接的学习问题——线性建模^①：在属性与响应之间学习线性关系。图 1-1 显示了从 1896 年开始，每次奥林匹克运动会（简称奥运会）男子 100 米比赛赢得金牌所需的比赛时间。我们的目标是用这些数据学习一个函数模型，此模型依赖于奥运会举办年份和 100 米获胜时间，并且用这个模型预测将来比赛中的获胜时间。显然，年份并不是影响获胜时间的唯一因素，如果我们认真对待这个预测，可能还会考虑其他因素（例如，主要参赛者的最近情况）。然而，通过图 1-1 可以看出，年份和获胜时间之间至少存在一个统计关系（它不可能是因果关系——时间的流逝并不是获胜时间下降的直接原因），并且这个例子足以帮助我们引入和发展线性建模的主要思想。

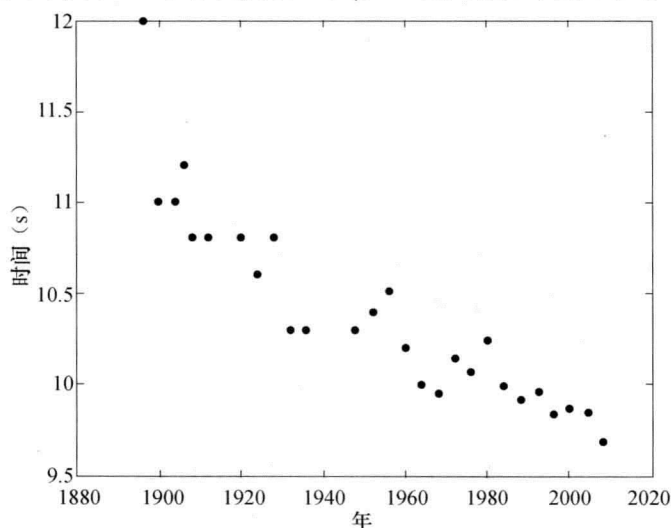


图 1-1 从 1896 年开始，夏季奥运会男子 100 米的获胜时间。注意：在 1914 年、1940 年和 1944 年，由于两次世界大战而中断了这个比赛

① 这里将要考虑的模型类型称为回归，它最初被 Francis Galton (1877 年) 用在遗传学方面。当时 Francis Galton 研究智力如何从一代传到一代（或者不是这样，由于这种情况也是可能的）。此术语后来被在统计背景下发展 Galton 工作的统计学家所采用。

1.1.1 定义模型

首先将模型定义为一个将输入属性（在这个例子中，是举办奥运会的年份）映射到输出或者目标值（获胜时间）的函数。对于属性，我们用年份的数值（如1980），尽管还有另外一个公式（例如，从第一届运动年开始， $1980 - 1896 = 84$ ），这对潜在的假设没有实质差别。

有许多函数可以定义这个映射。一般地，这个函数将以 x （奥运会年份）为输入，并且将返回 t （用秒表示的获胜时间）。也就是说， t 是 x 的函数。数学上，把这个记为 $t = f(x)$ 。在有些情况下，我们需要知道的是用来评估函数的 x 。例如，如果 $f(x) = \sin(x)$ ，或者 $f(x) = x$ ，那么对任何 x ，我们可以计算 t 。一般地，我们需要更灵活并且我们的模型可能有一个相关参数的集合。例如， $t = ax$ 有一个参数 a ，此参数需要用某种方法定义。在机器学习中，从一个合适的数据集中学习模型参数是一个普遍的问题。我们将用 $t = f(x; a)$ 来表示 x 与参数 a 之间的函数 $f(\cdot)$ 。

1.1.2 模型假设

为了便于选择特定的模型来使用，我们需要做一些假设。在这个阶段的初始假设是： x 与 t 之间的关系是线性的（参见注解1.1）。

注解 1.1（线性关系）：等式

$$y = mx + c$$

这里 m 和 c 是常量，在 x 和 y 之间定义了一个线性关系。它称为是线性的，因为从直观上看，在 x 与 y 之间的关系呈一条直线。下面的等式是非线性的，因为变量 x 和 y 的形式更复杂：

$$y = mx^2 + c^2, y = \sin(x), \sqrt{y} = mx + c$$

m 和 c 的值不影响关系的线性性。例如，如下都表示 x 和 y 之间的线性关系：

$$y = mx + c^2, y = x \sin(m) + c$$

或者可以表述为：

图 1-1 中的数据可以用一条直线模拟。

或者：

每 M 年，获胜时间下降相同数量。

观察图 1-1，我们可以看到这个假设并不是完全满足。然而，我们希望它是一个可用的模型，并且它可以对将来的获胜时间做出预测。

满足我们假设的最简单模型是

$$t = f(x) = x$$

获胜时间等于奥运会年份。 x 大于等于 1880， t 小于等于 12，随着年份的增长获胜时间在下降，这个事实说明这个模型是不适当的。添加一个单参数得到：

$$t = f(x; w) = wx$$

这里 w 为正或者负。这个改进的模型产生了一条直线，通过选择 w ，可以使这条直线有任何梯度。这个模型在灵活性（flexibility）方面有所提升，但是它仍然是受限制的，因为在奥运会年份 0 年时，模型预测的获胜时间是 $w \times 0 = 0$ 。通过这个数据可以看出，这是不现实的——按照数据的一般趋势，在 0 年时，获胜时间实际上应该是一个相当大的数。通过对模型添加多个参数，可以克服这个限制：

$$t = f(x; w_0, w_1) = w_0 + w_1 x \quad (1-1)$$

这是直线的标准等式，这个等式许多读者以前都遇到过。现在学习任务是用图 1-1 的数据为两个参数 w_0 和 w_1 选择合适的值。这两个参数常常认为是截距 (w_0 ，直线与 t 轴的截距) 和梯度 (w_1 ，直线的梯度)，以及改变它们的影响 (effect)，如图 1-2 所示 (MATLAB 脚本: plotlinear.m) (参见练习 EX 1.1)。

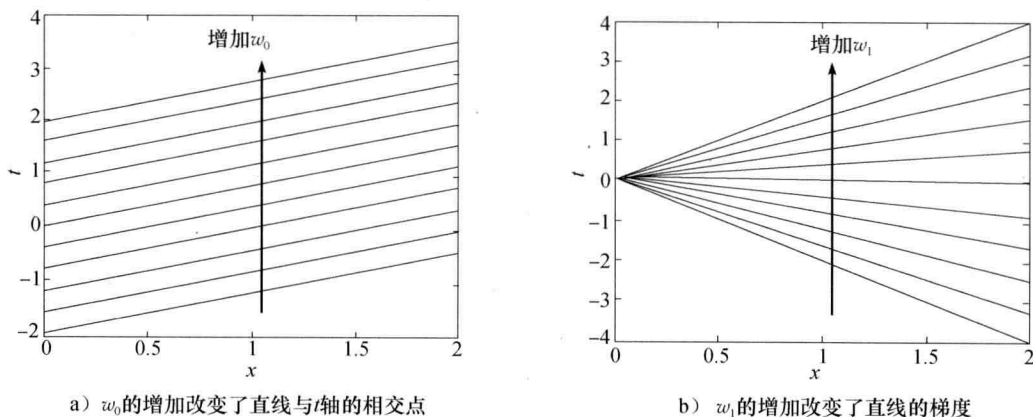


图 1-2 在式 (1-1) 定义的线性模型中，改变 w_0 和 w_1 带来的影响

1.1.3 定义什么是好的模型

为了选择在某种方式下最好的 w_0 和 w_1 值，我们需要定义最好的意义是什么。常识表明所谓最好的解是由 w_0 和 w_1 的一些值组成，这些值可以产生一条能尽可能与所有数据点接近的直线。衡量一个特定模型与数据点接近程度的普遍方法是真正的获胜时间与模型预测的获胜时间之间的平方差。用 x_n 、 t_n 分别表示第 n 次的奥运会年份和获胜时间，平方差定义为：

$$(t_n - f(x_n; w_0, w_1))^2$$

这个数值越小，模型在 x_n 处越接近 t_n 。对差值取平方是很重要的。如果不这样做，就可以通过连续增加 $f(x_n; w_0, w_1)$ 来无限减小这个量。

这个表达称为平方损失函数 (squared loss function)，因为它描述了使用 $f(x_n; w_0, w_1)$ 模拟 t_n 所损失的精度。在本章中，我们用 $\mathcal{L}(\cdot)$ 表示损失函数。在这种情况下，

$$\mathcal{L}_n(t_n, f(x_n; w_0, w_1)) = (t_n - f(x_n; w_0, w_1))^2 \quad (1-2)$$

是 n 年的损失。损失总是正的，并且损失越小，函数描述这个数据就越好。由于对于所有的 N 年，我们想有一个低的损失，所以考虑在整个数据集上的平均损失，即

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N \mathcal{L}_n(t_n, f(x_n; w_0, w_1)) \quad (1-3)$$

这是每 N 年的平均损失值。它越低越好。因此我们将调整 w_0 和 w_1 值来产生一个模型，此模型得到平均损失的最低值 \mathcal{L} 。寻找 w_0 和 w_1 的最好值，用数学表达式可以表示为

$$\arg \min_{w_0, w_1} \frac{1}{N} \sum_{n=1}^N \mathcal{L}_n(t_n, f(x_n; w_0, w_1))$$

argmin 项是数学上“找到最小化参数”的缩写。在这个例子中，参数是 w_0 和 w_1 的值同时

最小化的表达式是平均损失。图 1-3 显示了一个假设的损失, 它是单参数 w 的函数。使 \mathcal{L} 最小的参数 w 的值是 $w=5$ 。历史上, 平方损失的最小化是函数估计的最小二乘误差法的基础, 它是由 Gauss 和 Legendre (1809 年) 在预测行星运动时发展的方法。

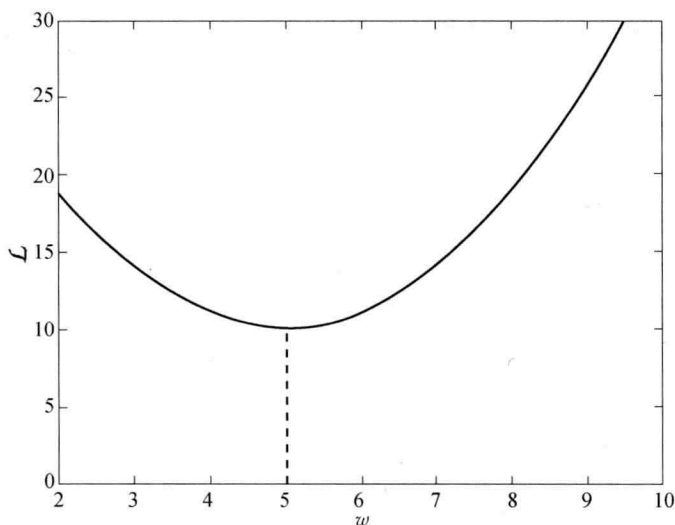


图 1-3 单参数 (w) 损失函数的例子。虚线表明了 $w=5$ 时损失最小

其他的损失函数适合回归。例如, 另一个常用的是绝对损失:

$$\mathcal{L}_n = |t_n - f(x_n; w_0, w_1)|$$

平方损失是非常常见的选择, 部分上由于它找到 w_0 和 w_1 的最好值相对直接这一事实——我们可以得到一个分析解。然而, 现代计算能力已经降低了数学方便的重要性——在多个适合的数据上选择一个方便的损失函数不再有任何借口。显然, 我们的目标是介绍对平方损失合适的通用模型概念。值得注意的是, 在许多情况下, 还有其他一些模型是可行的并且可能是更适合的。

1.1.4 最小二乘解: 一个有效的例子

简要说明我们的数据集由 $n=1, \dots, N$ 观测值构成, 它们中的每一个由一个年 x_n 和时间 (秒) t_n 构成。

我们继续尽力寻找一个函数关系, 此函数关系用一个线性模型定义为

$$f(x; w_0, w_1) = w_0 + w_1 x \quad (1-4)$$

我们决定将用最小二乘损失函数来选择适合的 w_0 和 w_1 。用表达式中的线性模型替代平均损失, 在括号外面相乘结果为

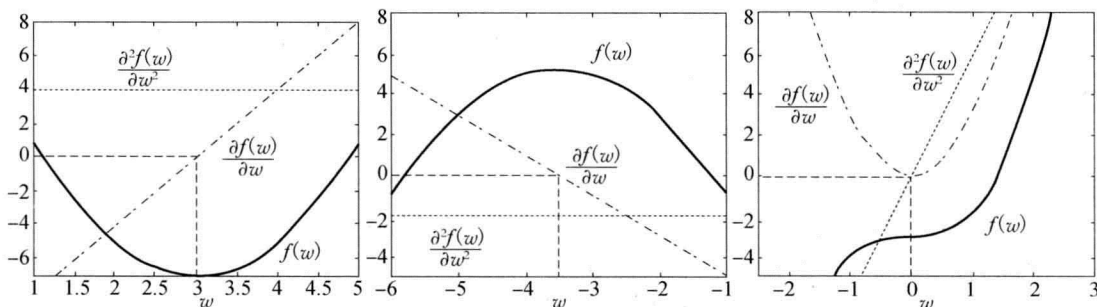
$$\begin{aligned} \mathcal{L} &= \frac{1}{N} \sum_{n=1}^N \mathcal{L}_n(t_n, f(x_n; w_0, w_1)) \\ &= \frac{1}{N} \sum_{n=1}^N (t_n - f(x_n; w_0, w_1))^2 \\ &= \frac{1}{N} \sum_{n=1}^N (t_n - (w_0 + w_1 x_n))^2 \\ &= \frac{1}{N} \sum_{n=1}^N (w_1^2 x_n^2 + 2w_1 x_n w_0 - 2w_1 x_n t_n + w_0^2 - 2w_0 t_n + t_n^2) \end{aligned}$$

$$= \frac{1}{N} \sum_{n=1}^N (w_1^2 x_n^2 + 2w_1 x_n (w_0 - t_n) + w_0^2 - 2w_0 t_n + t_n^2) \quad (1-5)$$

对损失函数求导数：在 \mathcal{L} 的最小值点处，其关于 w_0 和 w_1 的偏导数一定是 0（参见注解 1.2）。因此，求出偏导数，使其等于 0 并对 w_0 和 w_1 求解，解 w_0 和 w_1 可以使我们得到最小值。从 w_1 开始，我们知道在式 (1-5) 中不包含 w_1 的项可以被忽略（由于这些项关于 w_1 的偏导数为 0）。去掉这些项得到

$$\frac{1}{N} \sum_{n=1}^N [w_1^2 x_n^2 + 2w_1 x_n w_0 - 2w_1 x_n t_n]$$

注解 1.2 (拐点)：通过搜索使函数梯度 $\frac{\partial f(w)}{\partial w}$ 为 0 的点，可以找到函数 $f(w)$ 的拐点（可能对应于最小值）。为了确定一个拐点是最大值、最小值还是鞍点，需要检验其 2 阶导数 $-\frac{\partial^2 f(w)}{\partial w^2}$ 。在拐点 \hat{w} ，如果其 2 阶导数是正的，那么这个拐点是最低点。下面三个图显示了三个例子函数及其 1 阶和 2 阶导数：



一般地，一个函数可能有多个拐点。一个有趣的特殊情况是，如果函数的 2 阶导数是正的常量，那么这个函数仅有一个最低点。

在求偏导数之前，我们重新排列这个表达式，使它更简单。尤其是，把没有下标 n 的项放在和的外面并重新排列得到的结果

$$w_1^2 \frac{1}{N} \left(\sum_{n=1}^N x_n^2 \right) + 2w_1 \frac{1}{N} \left(\sum_{n=1}^N x_n (w_0 - t_n) \right)$$

如下表达式给出了其关于 w_1 的偏导数

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1 \frac{1}{N} \left(\sum_{n=1}^N x_n^2 \right) + \frac{2}{N} \left(\sum_{n=1}^N x_n (w_0 - t_n) \right) \quad (1-6)$$

现在对 w_0 做相同的操作。去掉不含 w_0 的项后，得到

$$\frac{1}{N} \sum_{n=1}^N [w_0^2 + 2w_1 x_n w_0 - 2w_0 t_n]$$

另外，我们在求导之前重新排列它。将没有下标 n 的项移到和的外面（注意 $\sum_{n=1}^N w_0^2 = Nw_0^2$ ），结果为

$$w_0^2 + 2w_0 w_1 \frac{1}{N} \left(\sum_{n=1}^N x_n \right) - 2w_0 \frac{1}{N} \left(\sum_{n=1}^N t_n \right)$$

对 w_0 求偏导数得

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{N} \left(\sum_{n=1}^N x_n \right) - \frac{2}{N} \left(\sum_{n=1}^N t_n \right) \quad (1-7)$$

导数等于0：现在我们有损失函数关于 w_0 和 w_1 的偏导数表达式。为了找到对应于拐点（希望是最小值点）的 w_0 和 w_1 值，必须使这些表达式为0并且对 w_0 和 w_1 求解。从关于 w_0 的表达式开始。将式（1-7）设置为0并且对 w_0 求解：

$$2w_0 + 2w_1 \frac{1}{N} \left(\sum_{n=1}^N x_n \right) - \frac{2}{N} \left(\sum_{n=1}^N t_n \right) = 0$$

$$2w_0 = \frac{2}{N} \left(\sum_{n=1}^N t_n \right) - w_1 \frac{2}{N} \left(\sum_{n=1}^N x_n \right)$$

$$w_0 = \frac{1}{N} \left(\sum_{n=1}^N t_n \right) - w_1 \frac{1}{N} \left(\sum_{n=1}^N x_n \right)$$

将平均获胜时间表示为 $\bar{t} = \frac{1}{N} \sum_{n=1}^N t_n$ 以及平均奥运会年份为 $\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n$ ，在拐点 \hat{w}_0 处，可以重写 w_0 值的表达式为

$$\hat{w}_0 = \bar{t} - w_1 \bar{x} \quad (1-8)$$

我们从这个表达式可以洞悉到什么？这个新的表达式是初始表示（ $t_n = w_0 + w_1 x_n$ ）的重新排列，这里 t_n 和 x_n 已经被平均值 \bar{t} 和 \bar{x} 取代。考虑在 N 个数据点上的平均函数值，表达式如下：

$$\frac{1}{N} \sum_{n=1}^N f(x_n; w_0, w_1) = \frac{1}{N} \sum_{n=1}^N (w_0 + w_1 x_n) = w_0 + w_1 \bar{x}$$

平均获胜时间通过 \bar{t} 给出，因此在式（1-8）中，选择 \hat{w}_0 来确保函数的平均值等于平均获胜时间。直观地，用这种方式匹配的平均值似乎是非常有意义的。

在我们用式（1-6）得到关于 \hat{w}_1 （ w_1 在拐点处的值，见注解 1.2）的表达式之前，值得简要地检验它的2阶导数，以确保它是最小值点。再一次对式（1-6）关于 w_1 求导并对式（1-7）关于 w_0 求导，结果为：

$$\begin{aligned} \frac{\partial^2 \mathcal{L}}{\partial w_1^2} &= \frac{2}{N} \sum_{n=1}^N x_n^2 \\ \frac{\partial^2 \mathcal{L}}{\partial w_0^2} &= 2 \end{aligned} \quad (1-9)$$

这两个量一定都是正的。这说明它仅有一个拐点并且此拐点对应于损失函数的最小值。

我们将此过程应用于关于 \hat{w}_0 （最小化损失函数的 w_0 的值）的表达式中。这个表达式依赖于 w_1 ，这暗示了对于特定的 w_1 ，我们知道最好的 w_0 。式（1-6）用我们的表达式替代最好的 w_0 （式（1-8））并重新排列，我们得到仅含 w_1 项的表达式：

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_1} &= w_1 \frac{2}{N} \left(\sum_{n=1}^N x_n^2 \right) + \frac{2}{N} \left(\sum_{n=1}^N x_n (\hat{w}_0 - t_n) \right) \\ &= w_1 \frac{2}{N} \left(\sum_{n=1}^N x_n^2 \right) + \frac{2}{N} \left(\sum_{n=1}^N x_n (\bar{t} - w_1 \bar{x} - t_n) \right) \\ &= w_1 \frac{2}{N} \left(\sum_{n=1}^N x_n^2 \right) + \bar{t} \frac{2}{N} \left(\sum_{n=1}^N x_n \right) - w_1 \bar{x} \frac{2}{N} \left(\sum_{n=1}^N x_n \right) - \frac{2}{N} \left(\sum_{n=1}^N x_n t_n \right) \end{aligned}$$

依然用 $\bar{x} = (1/N) \sum_{n=1}^N x_n$ 来简化这个表达式并合并包含 w_1 的项：

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1 \left[\left(\frac{1}{N} \left(\sum_{n=1}^N x_n^2 \right) \right) - \bar{x} \bar{x} \right] + 2\bar{t} \bar{x} - 2 \frac{1}{N} \left(\sum_{n=1}^N x_n t_n \right)$$

最后，通过把这个偏导数设置为0，我们能得到关于 \hat{w}_1 的表达式并且对 w_1 求解：

$$\begin{aligned}
2w_1 \left[\left(\frac{1}{N} \left(\sum_{n=1}^N x_n^2 \right) \right) - \bar{x} \bar{x} \right] + 2\bar{t} \bar{x} - 2 \frac{1}{N} \left(\sum_{n=1}^N x_n t_n \right) &= 0 \\
2w_1 \left[\left(\frac{1}{N} \left(\sum_{n=1}^N x_n^2 \right) \right) - \bar{x} \bar{x} \right] &= 2 \frac{1}{N} \left(\sum_{n=1}^N x_n t_n \right) - 2\bar{t} \bar{x} \\
\hat{w}_1 &= \frac{\frac{1}{N} \left(\sum_{n=1}^N x_n t_n \right) - \bar{t} \bar{x}}{\frac{1}{N} \left(\sum_{n=1}^N x_n^2 \right) - \bar{x} \bar{x}}
\end{aligned}$$

现在定义一些新的平均量是非常有用的。第一个， $(1/N) \sum_{n=1}^N x_n^2$ 是数据的平均平方值并且我们把它记为 $\overline{x^2}$ 。注意，这个量与 $(\bar{x})^2$ 不同。第二个是 $(1/N) \sum_{n=1}^N x_n t_n$ (同样，它与 $\bar{x} \bar{t}$ 不同)。我们将它记为 \overline{xt} 。将这些在关于 w_1 的表达式中替换，得到：

$$\hat{w}_1 = \frac{\overline{xt} - \bar{x} \bar{t}}{\overline{x^2} - (\bar{x})^2} \quad (1-10)$$

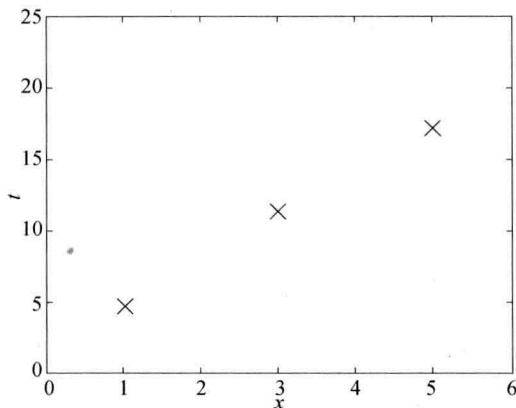
式 (1-10) 和式 (1-8) 为计算最好的参数值提供了全部所需的一切。首先用式 (1-10) 的 \hat{w}_1 替换式 (1-8) 来计算 \hat{w}_0 (MATLAB 脚本: fitlinear.m)。

1.1.5 有效的例子

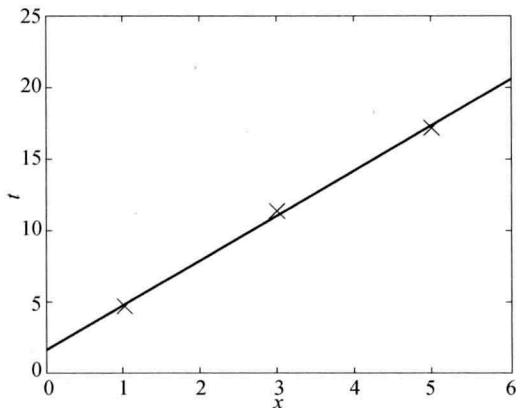
在用线性模型拟合奥运会数据之前，在一个更小数据集上提供一个有效的例子是非常有用的。假设我们观察到 $N=3$ 个数据点，如表 1-1 所示。最后一行给出了计算 \hat{w}_0 和 \hat{w}_1 所需的各种平均值: \bar{x} 、 \bar{t} 、 \overline{xt} 和 $\overline{x^2}$ 。图 1-4 画出了这 3 个数据点。

表 1-1 线性回归例子的合成数据集

n	x_n	t_n	$x_n t_n$	x_n^2
1	1	4.8	4.8	1
2	3	11.3	33.9	9
3	5	17.2	86	25
$(1/N) \sum_{n=1}^N$	3	11.1	41.57	11.67



a) 在表1-1中描述的3个合成数据点



b) 由 $f(x; w_0, w_1) = 1.8 + 3.1x$ 定义的最小二乘拟合

图 1-4 1.1.5 节中有效的例子中的数据和函数

将这些值代入式 (1-10)，得到：

$$\begin{aligned}w_1 &= \frac{41.57 - 3 \times 11.1}{11.67 - 3 \times 3} \\&= \frac{8.27}{2.67} \\&= 3.1\end{aligned}$$

和

$$w_0 = 11.1 - 3.1 \times 3 = 1.8$$

因此最好的线性函数是：

$$f(x;w_0,w_1) = 1.8 + 3.1x$$

并且如图 1-4b 所示。

1. 1. 6 奥运会数据的最小二乘拟合

表 1-2 总结了奥运会 100 米数据集的数据（见图 1-1）。

表 1-2 奥运会男子 100 米数据

n	x_n	t_n	$x_n t_n$	x_n^2
1	1896	12.00	22 752.0	3.5948×10^6
2	1900	11.00	20 900.0	3.6100×10^6
3	1904	11.00	20 944.0	3.6252×10^6
4	1906	11.20	21 347.2	3.6328×10^6
5	1908	10.80	20 606.4	3.6405×10^6
6	1912	10.80	20 649.6	3.6557×10^6
7	1920	10.80	20 736.0	3.6864×10^6
8	1924	10.60	20 394.4	3.7018×10^6
9	1928	10.80	20 822.4	3.7172×10^6
10	1932	10.30	19 899.6	3.7326×10^6
11	1936	10.30	19 940.8	3.7481×10^6
12	1948	10.30	20 064.4	3.7947×10^6
13	1952	10.40	20 300.8	3.8103×10^6
14	1956	10.50	20 538.0	3.8259×10^6
15	1960	10.20	19 992.0	3.8416×10^6
16	1964	10.00	19 640.0	3.8573×10^6
17	1968	9.95	19 581.6	3.8730×10^6
18	1972	10.14	19 996.1	3.8888×10^6
19	1976	10.06	19 878.6	3.9046×10^6
20	1980	10.25	20 295.0	3.9204×10^6
21	1984	9.99	19 820.2	3.9363×10^6
22	1988	9.92	19 721.0	3.9521×10^6
23	1992	9.96	19 840.3	3.9681×10^6
24	1996	9.84	19 640.6	3.9840×10^6
25	2000	9.87	19 740.0	4.0000×10^6
26	2004	9.85	19 739.4	4.0160×10^6
27	2008	9.69	19 457.5	4.0321×10^6
$(1/N) \sum_{n=1}^N$	1952.37	10.39	20 268.1	3.8130×10^6

将相同的方法精确地应用到这个数据，对 w_0 和 w_1 得到如下值（注意，最后值用 MATLAB 计算——如果你计算完成，由于舍入误差，可能得到的值略有些不同）：

$$\begin{aligned} w_1 &= \frac{20\,268.1 - 1952.37 \times 10.39}{3.8130 \times 10^6 - 1952.37 \times 1952.37} \\ &= \frac{-16.3}{1225.5} \\ &= -0.0133 \\ w_0 &= 10.39 - (-0.0133) \times 1952.37 \\ &= 36.416 \end{aligned}$$

因此，最好的线性函数是：

$$f(x; w_0, w_1) = 36.416 - 0.013x \quad (1-11)$$

图 1-5 画出了此函数（参见练习 EX 1.2）。这些值和练习 EX 1.1 得到的估计值一致吗？（MATLAB 脚本：fitolympic.m。）

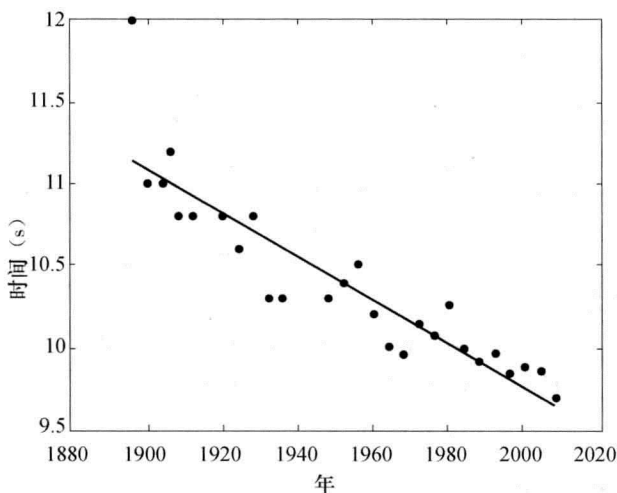


图 1-5 对奥运会男子 100 米数据集的最小二乘拟合 ($f(x; w_0, w_1) = 36.416 - 0.013x$)

1.1.7 小结

到目前为止，我们介绍了创建一个总结属性集合和响应集合之间关系模型（尤其是线性模型）的思想。为了从数据中拟合（或者学习）这个模型，我们定义了一个损失函数来客观地评价一个特定模型的好坏程度。用平方损失，我们得到了最小化损失模型参数值的精确表达式，并且因此得到了最好的函数。最后，我们把这个技术应用到两个不同的数据集。现在我们将看到怎样用模型做出预测。

1.2 预测

现在我们有了一个将奥运会年份和 100 米短跑比赛的获胜时间联系起来的模型，我们能用它对某年还没观察到的获胜时间做出预测。例如，预测 2012 年和 2016 年奥运会的获胜时间 t^{2012} 和 t^{2016} ，我们将 $x=2012$ 和 $x=2016$ 代入公式：

$$\begin{aligned} f(x; w_0 = 36.416, w_1 = -0.0133) &= 36.416 - 0.0133x \\ t^{2012} = f(2012; w_0, w_1) &= 36.416 - 0.0133 \times 2012 = 9.595 \end{aligned}$$

$$t^{2016} = f(2016; w_0, w_1) = 36.416 - 0.0133 \times 2016 = 9.541$$

这些预测在图 1-6 中给出 (MATLAB 脚本: olymppred.m)。从图 1-6 中可以看出, 基于线性回归模型, 我们预期 2012 年伦敦奥运会的获胜时间是 9.595s。这个值是非常精确的。任何模型对如此复杂的事件预测出如此高精度的结果似乎是不可能的, 仅仅是基于直线。我们的模型看起来是非常精确的, 但是对于某些数据还是不能做出预测, 从图 1-5 中的直线到某些点的距离也可以看出这点。对将来的预测是更准确的这一假设似乎是特别愚蠢的。

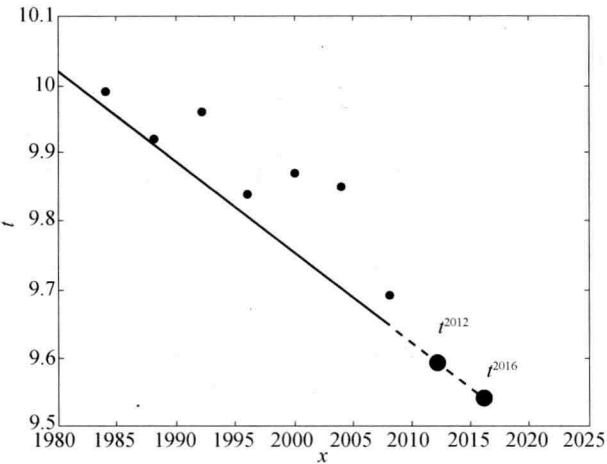


图 1-6 从 1980 年开始奥运会男子 100 米短跑的获胜时间放大图, 显示了对 2012 年和 2016 年奥运会的预测

准确的预测在模型不完美的情况下 (几乎所有情况) 是有限的。一般地, 在一个范围内的值它是有用的, 而不是任何一个特定值。在第 2 章及之后章节, 我们将看到如何完成这些预测。

9
12

1.2.1 第二个奥运会数据集

表 1-3 显示了第二个数据集, 它和第一个数据集是相关的。图 1-7 显示了此数据集及其最小化平方损失函数的线性模型 (参见练习 EX 1.6 和练习 EX 1.7)。女子数据的模型是 (注意, 由于舍入误差, 与这些数据相比, 可能产生一些差异):

表 1-3 奥运会女子 100 米数据

n	x_n	t_n	$x_n t_n$	x_n^2
1	1928	12.20	23 521.6	3.7172×10^6
2	1932	11.90	22 990.8	3.7326×10^6
3	1936	11.50	22 264.0	3.7481×10^6
4	1948	11.90	23 181.2	3.7947×10^6
5	1952	11.50	22 448.0	3.8103×10^6
6	1956	11.50	22 494.0	3.8259×10^6
7	1960	11.00	21 560.0	3.8416×10^6
8	1964	11.40	22 389.6	3.8573×10^6
9	1968	11.00	21 648.0	3.8730×10^6

(续)

n	x_n	t_n	$x_n t_n$	x_n^2
10	1972	11.07	21 830.0	3.8888×10^6
11	1976	11.08	21 894.1	3.9046×10^6
12	1980	11.06	21 898.8	3.9204×10^6
13	1984	10.97	21 764.5	3.9363×10^6
14	1988	10.54	20 953.5	3.9521×10^6
15	1992	10.82	21 553.4	3.9681×10^6
16	1996	10.94	21 836.2	3.9840×10^6
17	2000	11.12	22 240.0	4.0000×10^6
18	2004	10.93	21 903.7	4.0160×10^6
19	2008	10.78	21 646.2	4.0321×10^6
$(1/N) \sum_{n=1}^N$	1970.74	11.22	22 106.2	3.8844×10^6

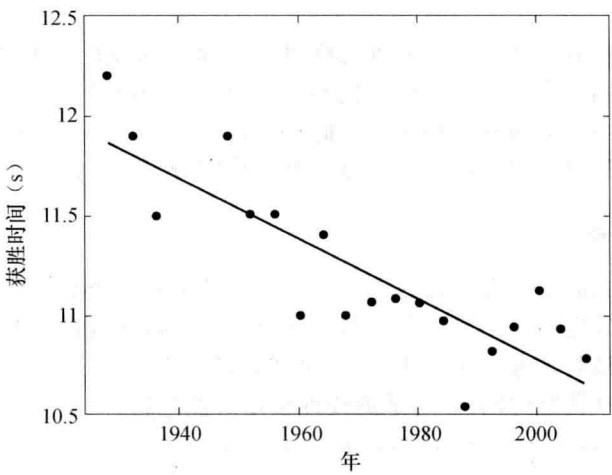


图 1-7 奥运会女子 100 米数据及其最小化平方损失的线性模型

$$f(x;w_0,w_1) = 40.92 - 0.015x$$

这个模型与从男子数据得到的模型相比是非常有趣的：

$$f(x;w_0,w_1) = 36.416 - 0.013x$$

女子模型有一个更高的截距 (w_0) 以及一个更陡峭的负梯度 (w_1)。如果将这两个模型放到一起，参见图 1-8，从中可以看到更高的截距以及更大的负梯度，这意味着在某点这两条直线将相交。用这个模型我们可以预测女子获胜时间比男子获胜时间快的第一届奥运会年份。根据得到的模型这将是在 2592 年奥运会（实际答案被舍入到最近的奥运会年份，并且已经通过 MATLAB 用精确数据计算得出，因此你可能发现一些微小的舍入误差（参见练习 EX 1.8））。

由于采用单个模型的点预测，所以从这个预测产生的置信区间不应该太大。不仅预测精度是可疑的，而且到最后的观测数据点将是很长的时间。我们能否假设获胜时间与奥运会年份的关系在将来一直按这个继续下去？如果这个假设成立，那么最后将会出现获胜时间 0 秒，这显然是不可能的。

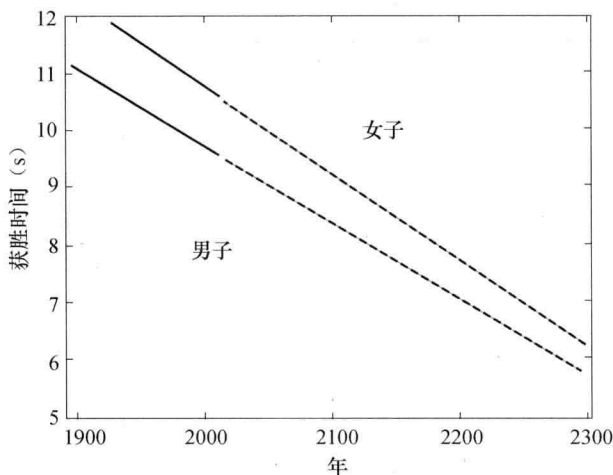


图 1-8 男子和女子函数对将来的预测

1.2.2 小结

通过前面介绍的内容，我们已经看到怎样将一个简单的线性模型拟合到一个小数据集，并用结果模型做出预测。我们描述了这种预测方式的一些局限性，在之后的章节中，我们将介绍另一种技术，它可以克服这些局限性。此时，特征 (x_n) 一直是单个数字。下面将看到线性模型怎样被延伸到更大的属性集合，这也促使我们对更复杂的关系建立模型。

1.3 向量/矩阵符号

在许多应用中，我们感兴趣的是这样一些问题：其中每一个数据点表示为一些属性的集合。例如，我们可以确定仅仅用奥运会年份不适合奥运会短跑数据模型的建立。用奥运会年份和每个运动员个人最好成绩建立的模型可能更准确。用 s_1, s_2, \dots, s_8 表示在跑道 1~8 的运动员的最好成绩（获胜时间），合适的线性模型可能包括：

$$t = f(x, s_1, \dots, s_8; w_0, \dots, w_9) = w_0 + w_1 x + w_2 s_1 + w_3 s_2 + w_4 s_3 + w_5 s_4 + w_6 s_5 + w_7 s_6 + w_8 s_7 + w_9 s_8$$

我们可以执行一遍以前的分析来找到 $\hat{w}_0, \dots, \hat{w}_9$ 。求得损失函数的偏导数之后，得到 10 个等式，它们再经过重新排序和相互替换。这是费时的练习，并且随着包含变量的进一步增加，它们很快变得不可行——具有数千个变量的机器学习是常见的。幸运的是，这里有另一种方法——用向量和矩阵。

由于有些读者可能不熟悉这个领域，所以我们将花一些时间来描述向量和矩阵的概念，以及数学上怎样将一些量处理成向量和矩阵的形式。对这些概念熟悉的读者可以直接学习 1.4 节。

通过将多个属性堆放在一起形成一个向量，可以将每个数据点的 9 个属性（8 个人最好成绩和一个奥运会年份）结合成一个单独的变量。我们将用黑体小写字母标记这些向量，例如 \mathbf{x}_n （参见注解 1.3）。我们常常需要提到一个特定向量或者矩阵的单个元素，这里用下标清晰地表示所提到的元素。例如，向量 \mathbf{x}_n 的第一个元素记为 x_{n1} ，第 i 个元素为 x_{ni} 。

注解 1.3 (标量、向量和矩阵)：我们将遵循表示的标准习惯，用字母（例如 x ）来表示标量，用黑体小写字母（例如 \mathbf{x} ）表示向量，用黑体大写字母（例如 \mathbf{X} ）表示矩阵。同时我们将始终坚持这一表示法，不同地方对向量的定义有不同的方式。例如， \bar{x} 是向量 \mathbf{x} 的普遍写法。

如果想在向量中表示所有的元素，我们将它以列的形式写出来，并用中括号括起来。这里是长度为2和4的两个向量的例子：

$$\mathbf{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

由于将向量记为列有点笨拙，所以我们常常将它们记为行，并且用转置符（参见注解1.4）来表示它们应该被旋转过来。如果我们假设有 D 个属性，那么将 \mathbf{x}_n 定义为 $\mathbf{x}_n = [x_{n1}, \dots, x_{nD}]$ 。在我们的奥运会数据中， $\mathbf{x} = [\text{Year}, s_1, s_2, \dots, s_8]^T$ 。

注解 1.4 (向量转置)：向量 \mathbf{x} 的转置（记为 \mathbf{x}^T ）是通过旋转此向量得到的，它是一行多列而不是一列多行。例如：

$$\mathbf{x} = \begin{bmatrix} 4 \\ 7 \\ 11 \\ -2 \end{bmatrix}, \mathbf{x}^T = [4, 7, 11, -2]$$

注解 1.5 (矩阵/向量维数和标引)：如果我们要引用一个矩阵或者向量的大小（或者维数），我们就给出两个数，并以行数开始。例如，

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

维数是 3×2 。向量是第二维为1的矩阵的一种特殊情况。例如，

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

可以被想象成维数为 4×1 的矩阵。

当在一个向量中标引其元素时，单个数就足够了（例如， y_3 记为上面 \mathbf{y} 向量的第三个元素）。当标引一个矩阵时，我们将用从行开始的两个下标。例如， a_{21} 表示 \mathbf{A} （上面）在第二行第一列的元素。注意，有时也用一个下标标记某个对象索引。例如， \mathbf{x}_n 是第 n 个属性集的向量。如果存在，这个索引总是首先呈现出来的。从上下文中显然可以看出这个索引是否存在。

在我们着手添加额外的变量时，有必要以向量形式重复对初始模型（ $t = w_0 + w_1 x$ ）的分析。这需要对在两种情况下得到的 \hat{w}_0 和 \hat{w}_1 的表达式进行比较。第一步，将 w_0 和 w_1 合并为单个参数向量 \mathbf{w} ，并将每个 x_n 扩大为1，从而产生数据向量 \mathbf{x}_n ，即

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \mathbf{x}_n = \begin{bmatrix} 1 \\ x_n \end{bmatrix}$$

依据 \mathbf{x}_n 和 \mathbf{w} ，此模型可以表示为（矩阵/向量乘法将在注解1.7中定义）：

$$f(x_n; w_0, w_1) = \mathbf{w}^T \mathbf{x}_n = w_0 + w_1 x_n$$

我们可以用 $\mathbf{w}^T \mathbf{x}$ 替换 $w_0 + w_1 x$ 的任何一个实例。例如，平方损失 \mathcal{L} 可以表示为

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N (t_n - \mathbf{w}^T \mathbf{x}_n)^2 \quad (1-12)$$

事实上，很容易将平均损失表达为如下的函数形式，它是向量和矩阵的函数：

$$\mathcal{L} = \frac{1}{N} (\mathbf{t} - \mathbf{X}\mathbf{w})^T (\mathbf{t} - \mathbf{X}\mathbf{w})$$

为了理解它是怎么等于式 (1-12) 的，我们首先将所有 \mathbf{x}_n 合并为一个矩阵 \mathbf{X} ，并且将所有 t_n 合并为一个向量 \mathbf{t} ：

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}, \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}$$

注解 1.6 (矩阵转置)：对于每个矩阵 \mathbf{X} ，其转置 \mathbf{X}^T 是通过将其每行变为列以及每列变为行而形成的。例如，如果 $\mathbf{Y} = \mathbf{X}^T$ ，那么 $Y_{ij} = X_{ji}$ 。

$$\mathbf{X} = \begin{bmatrix} 1 & 4 \\ 3 & 6 \\ -2 & 11 \end{bmatrix}, \mathbf{X}^T = \begin{bmatrix} 1 & 3 & -2 \\ 4 & 6 & 11 \end{bmatrix}$$

注解 1.7 (矩阵乘法)：为了继续进行，我们必须引入矩阵乘法概念。 $N \times M$ 矩阵 \mathbf{A} 和 $P \times Q$ 矩阵 \mathbf{B} 进行乘法 \mathbf{AB} ，只有在 $M=P$ 时才有意义，即 \mathbf{A} 的列数和 \mathbf{B} 的行数相等。假设此条件成立，那么乘积 $\mathbf{C} = \mathbf{AB}$ 是一个 $N \times Q$ 矩阵，满足

$$C_{ij} = \sum_k A_{ik} B_{kj}$$

写出此矩阵常常是有用的，例如，

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \end{bmatrix}$$

这里我们可以想象，通过同时遍历 \mathbf{A} 的相关行和 \mathbf{B} 的相关列来计算 \mathbf{C} 的每个元素。

我们经常遇到的特殊情况是两个列向量之间的内积，定义为 $z = \mathbf{x}^T \mathbf{y}$ ，结果是一个标量。这两个向量必须是相同的长度，并且转置确保 \mathbf{x} 的列数和 \mathbf{y} 的行数相同。应用与矩阵一样的技术，我们看到

$$z = \sum_k x_k y_k$$

因此，如果执行矩阵乘法 $\mathbf{X}\mathbf{w}$ ，那么结果是一个向量，此向量的形式如下：

$$\mathbf{X}\mathbf{w} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \times \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} w_0 + w_1 x_1 \\ w_0 + w_1 x_2 \\ \vdots \\ w_0 + w_1 x_N \end{bmatrix}$$

\mathbf{t} 减去它得到：

$$\mathbf{t} - \mathbf{X}\mathbf{w} = \begin{bmatrix} t_1 - w_0 - w_1 x_1 \\ t_2 - w_0 - w_1 x_2 \\ \vdots \\ t_N - w_0 - w_1 x_N \end{bmatrix}$$

与其转置相乘，然后变换为平方和，从而得到我们原来的损失函数：

$$\begin{aligned}
 (\mathbf{X}\mathbf{w} - \mathbf{t})^T (\mathbf{X}\mathbf{w} - \mathbf{t}) &= (w_0 + w_1 x_1 - t_1)^2 + (w_0 + w_1 x_2 - t_2)^2 + \cdots \\
 &\quad + (w_0 + w_1 x_N - t_N)^2 \\
 &= \sum_{n=1}^N (w_0 + w_1 x_n - t_n)^2 \\
 &= \sum_{n=1}^N (t_n - f(x_n; w_0, w_1))^2
 \end{aligned}$$

因此，损失可以简洁地表示为：

$$\mathcal{L} = \frac{1}{N} (\mathbf{t} - \mathbf{X}\mathbf{w})^T (\mathbf{t} - \mathbf{X}\mathbf{w}) \quad (1-13)$$

下面的损失表达式都是相同的：

$$\mathcal{L} = \frac{1}{N} (\mathbf{t} - \mathbf{X}\mathbf{w})^T (\mathbf{t} - \mathbf{X}\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N (t_n - \mathbf{w}^T \mathbf{x}_n)^2 = \frac{1}{N} \sum_{n=1}^N (t_n - (w_0 + w_1 x_n))^2$$

注解 1.8 (乘积的转置)：矩阵乘积的转置 $(\mathbf{X}\mathbf{w})^T$ 可以将两项交换相乘并对两个单独的矩阵分别转置

$$(\mathbf{X}\mathbf{w})^T = \mathbf{w}^T \mathbf{X}^T$$

为了处理更复杂的形式，可以多次应用上面的结果，例如，

$$\begin{aligned}
 (\mathbf{ABCD})^T &= ((\mathbf{AB})(\mathbf{CD}))^T \\
 &= (\mathbf{CD})^T (\mathbf{AB})^T \\
 &= \mathbf{D}^T \mathbf{C}^T \mathbf{B}^T \mathbf{A}^T
 \end{aligned}$$

一旦添加了括号，就很容易产生矩阵损失。注意，矩阵相乘的顺序（注解 1.7 中给出了不同大小约束下的讨论）以及注解 1.8 中给出的乘积转置的定义都很重要：

$$\begin{aligned}
 \mathcal{L} &= \frac{1}{N} (\mathbf{X}\mathbf{w} - \mathbf{t})^T (\mathbf{X}\mathbf{w} - \mathbf{t}) \\
 &= ((\mathbf{X}\mathbf{w})^T - \mathbf{t}^T) (\mathbf{X}\mathbf{w} - \mathbf{t}) \\
 &= \frac{1}{N} (\mathbf{X}\mathbf{w})^T \mathbf{X}\mathbf{w} - \frac{1}{N} \mathbf{t}^T \mathbf{X}\mathbf{w} - \frac{1}{N} (\mathbf{X}\mathbf{w})^T \mathbf{t} + \frac{1}{N} \mathbf{t}^T \mathbf{t} \\
 &= \frac{1}{N} \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} - \frac{2}{N} \mathbf{w}^T \mathbf{X}^T \mathbf{t} + \frac{1}{N} \mathbf{t}^T \mathbf{t}
 \end{aligned} \quad (1-14)$$

$\mathbf{t}^T \mathbf{X}\mathbf{w}$ 和 $\mathbf{w}^T \mathbf{X}^T \mathbf{t}$ 互为转置（根据乘积转置的一致性）并且都是标量（各自满足其结果是 1×1 矩阵，因此是一个标量）。这意味着它们一定是相同的并且可以合并。

向量/矩阵的微分损失：我们想要和 \mathcal{L} 的一个拐点（极小值）一致的向量 \mathbf{w} 的值。为了实现它，必须求得 \mathcal{L} 关于 \mathbf{w} 的偏导数。依次获得 \mathcal{L} 关于 \mathbf{w} 每个元素的偏导数，将结果组成一个向量。尽管在后面可以看到，实际上能够直接获得 $\frac{\partial \mathcal{L}}{\partial \mathbf{w}}$ 的向量形式，但在这个例子中还是值得这样做的。以两个变量为例，向量表示为

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial w_0} \\ \frac{\partial \mathcal{L}}{\partial w_1} \end{bmatrix}$$

该向量包含 \mathcal{L} 关于 w_0 和 w_1 的偏导数。向量的这两个元素依次与式 (1-7) 和式 (1-6) 中的元素是一样的。通过运算这两个参数的微分方程式 (1-13)，能够检验我们的损失的确是正

确的。首先，展开表达式

$$\mathcal{L} = \frac{1}{N}(\mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} - 2 \mathbf{w}^T \mathbf{X}^T \mathbf{t} + \mathbf{t}^T \mathbf{t})$$

最后一项不包含 w_0 或 w_1 ，因此可以忽略。展开后，第一项是（见练习 EX 1.3）

$$w_0^2 \frac{1}{N} \left(\sum_{n=1}^N X_{n0}^2 \right) + 2w_0 w_1 \frac{1}{N} \left(\sum_{n=1}^N X_{n0} X_{n1} \right) + w_1^2 \frac{1}{N} \left(\sum_{n=1}^N X_{n1}^2 \right)$$

其中 X_{n0} 是 \mathbf{X} 第 n 行的第一个元素，即第 n 个数据对象的第一个元素， X_{n1} 是第二个（角标从 0 开始是为了与 w_0 一致）。类似地，第二项为

$$2w_0 \frac{1}{N} \left(\sum_{n=1}^N x_{n0} t_n \right) + 2w_1 \frac{1}{N} \left(\sum_{n=1}^N x_{n1} t_n \right)$$

结合这些以及在之前的表示法中， $X_{n0}=1$ ， $X_{n1}=x_n$ ，其结果为

$$w_0^2 + 2w_0 w_1 \frac{1}{N} \left(\sum_{n=1}^N x_n \right) + w_1^2 \frac{1}{N} \left(\sum_{n=1}^N x_n^2 \right) - 2w_0 \frac{1}{N} \left(\sum_{n=1}^N t_n \right) - 2w_1 \frac{1}{N} \left(\sum_{n=1}^N x_n t_n \right)$$

关于 w_0 和 w_1 均值和微分的简洁表示，可以表示为

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \bar{x} - 2\bar{t}$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_0 \bar{x} + 2w_1 \overline{x^2} - 2\overline{xt}$$

作为非正式的练习，请证明这些等价于从非矢量化损失函数（式（1-7）和式（1-6））获得的导数。

20

幸运的是，有很多标准的恒等式可以直接微分矢量化表达式。表 1-4 给出了需要的等式。

这些恒等式让其导数等于 0，可以直接得到下面的表 1-4 关于向量微分的一些有用等式

表式：

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} &= \frac{2}{N} \mathbf{X}^T \mathbf{X} \mathbf{w} - \frac{2}{N} \mathbf{X}^T \mathbf{t} = 0 \\ \mathbf{X}^T \mathbf{X} \mathbf{w} &= \mathbf{X}^T \mathbf{t} \end{aligned} \quad (1-15)$$

$f(\mathbf{w})$	$\frac{\partial f}{\partial \mathbf{w}}$
$\mathbf{w}^T \mathbf{x}$	\mathbf{x}
$\mathbf{x}^T \mathbf{w}$	\mathbf{x}
$\mathbf{w}^T \mathbf{w}$	$2 \mathbf{w}$
$\mathbf{w}^T \mathbf{C} \mathbf{w}$	$2 \mathbf{C} \mathbf{w}$

推导表达式 $\hat{\mathbf{w}}$ 的最后一步（ \mathbf{w} 的最优值）在式（1-15）中给出。我们不能两边都除以 $\mathbf{X}^T \mathbf{X}$ （除法在矩阵中没有定义），但是可以乘以一个矩阵以消除左边的 $\mathbf{X}^T \mathbf{X}$ （只留下一个单位矩阵，见注解 1.9）。要乘的矩阵是 $\mathbf{X}^T \mathbf{X}$ 的逆矩阵（见注解 1.10），表示为 $(\mathbf{X}^T \mathbf{X})^{-1}$ 。给式（1-15）前面乘 $(\mathbf{X}^T \mathbf{X})^{-1}$ ，得到：

$$\mathbf{I} \mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}$$

因为 $\mathbf{I} \mathbf{w} = \mathbf{w}$ （从单位矩阵的定义可知），所以我们得到使损失最小的 \mathbf{w} 值， $\hat{\mathbf{w}}$ 的矩阵公式：

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t} \quad (1-16)$$

注解 1.9（单位矩阵）：我们经常遇到单位矩阵 \mathbf{I}_N 。它是一个 $N \times N$ 矩阵，对角线全为 1，其他都为 0。

$$\mathbf{I}_1 = 1, \mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

通常，单位矩阵的大小可以很明显地从其表达式中看出。在这些例子中，我们忽略其下标。

单位矩阵的一个重要性质是任何一个向量或者矩阵与一个合适大小的单位矩阵相乘都等于该矩阵或者向量本身。例如，给定 $\mathbf{y} = [y_1, \dots, y_D]^T$ 以及一个 $D \times D$ 单位矩阵 \mathbf{I}_D ，则有

$$\mathbf{y}^T \mathbf{I}_D = \mathbf{y}, \mathbf{I}_D \mathbf{y} = \mathbf{y}$$

类似地，对于 $N \times M$ 矩阵，有

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NM} \end{bmatrix}$$

$$\mathbf{A} \mathbf{I}_M = \mathbf{A}, \mathbf{I}_N \mathbf{A} = \mathbf{A}$$

一个单位矩阵乘以一个标量，其结果是矩阵的每个对角线元素乘以该标量。从下面这个例子可以看出：

$$\sigma^2 \mathbf{I}_M = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix}$$

1.3.1 例子

通过公式展开检查得出，矩阵公式确实与之前得到的标量公式一致。在二维空间中，

$$\mathbf{X}^T \mathbf{X} = \begin{bmatrix} \sum_{n=1}^N x_{n0}^2 & \sum_{n=1}^N x_{n0} x_{n1} \\ \sum_{n=1}^N x_{n1} x_{n0} & \sum_{n=1}^N x_{n1}^2 \end{bmatrix}$$

\bar{x} 表示均值，上式重写为

$$\mathbf{X}^T \mathbf{X} = N \begin{bmatrix} \overline{x_0^2} & \overline{x_0 x_1} \\ \overline{x_1 x_0} & \overline{x_1^2} \end{bmatrix}$$

2×2 矩阵的逆（见注解 1.10）可以表示为

注解 1.10 (矩阵的逆)：矩阵 \mathbf{A} 的逆定义为矩阵 \mathbf{A}^{-1} ，满足 $\mathbf{A}^{-1} \mathbf{A} = \mathbf{I}$ 。在此没有给出矩阵求逆的一般形式，但是从数学角度看，一个 2×2 矩阵可以用以下公式求逆：

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \mathbf{A}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

我们经常会遇到对角矩阵求逆的特殊情况（即对角线以外的元素都是 0）。这样矩阵的逆是另一个对角矩阵，它的每个对角线元素仅仅是原矩阵相应位置元素的逆。例如，

$$\mathbf{A} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{DD} \end{bmatrix}, \mathbf{A}^{-1} = \begin{bmatrix} a_{11}^{-1} & 0 & \cdots & 0 \\ 0 & a_{22}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{DD}^{-1} \end{bmatrix}$$

值得注意的是，这个定义表明一个单位矩阵（见注解 1.9）的逆仍然是一个单位矩阵：

$$\mathbf{I}^{-1} = \mathbf{I}$$

$$(\mathbf{X}^T \mathbf{X})^{-1} = \frac{1}{N} \frac{1}{\overline{x_0^2} \overline{x_1^2} - \overline{x_1 x_0} \overline{x_0 x_1}} \begin{bmatrix} \overline{x_1^2} & -\overline{x_0 x_1} \\ -\overline{x_1 x_0} & \overline{x_0^2} \end{bmatrix}$$

我们需要给该式乘以 $\mathbf{X}^T \mathbf{t}$, 即 (直接用平均值标记)

$$N \begin{bmatrix} \overline{x_0 t} \\ \overline{x_1 t} \end{bmatrix}$$

现在, 知道 x_{n0} 一直为 1, x_{n1} 重新定义为 x_n (和标量标记一致), 我们需要计算:

$$\hat{\mathbf{w}} = \frac{1}{N} \frac{1}{\overline{x^2} - \overline{x} \overline{x}} \begin{bmatrix} \overline{x^2} & -\overline{x} \\ -\overline{x} & 1 \end{bmatrix} \times N \begin{bmatrix} \overline{t} \\ \overline{xt} \end{bmatrix}$$

即

$$\hat{\mathbf{w}} = \begin{bmatrix} \hat{w}_0 \\ \hat{w}_1 \end{bmatrix} = \left(\frac{1}{\overline{x^2} - \overline{x} \overline{x}} \right) \begin{bmatrix} \overline{x^2} \overline{t} - \overline{x} \overline{xt} \\ -\overline{x} \overline{t} + \overline{xt} \end{bmatrix} \quad (1-17)$$

以 \hat{w}_1 开始 (第二行)

$$\hat{w}_1 = \frac{\overline{xt} - \overline{x} \overline{t}}{\overline{x^2} - \overline{x} \overline{x}}$$

和前面完全一样, \hat{w}_0 需要简单重排, 从而使后向计算更容易。从原表达式开始, 将 \hat{w}_1 的新表达式代入

$$\begin{aligned} \hat{w}_0 &= \overline{t} - \hat{w}_1 \overline{x} \\ &= \overline{t} - \overline{x} \frac{\overline{xt} - \overline{x} \overline{t}}{\overline{x^2} - \overline{x} \overline{x}} \\ &= \overline{t} \left(\frac{\overline{x^2} - \overline{x} \overline{x}}{\overline{x^2} - \overline{x} \overline{x}} \right) - \overline{x} \frac{\overline{xt} - \overline{x} \overline{t}}{\overline{x^2} - \overline{x} \overline{x}} \\ &= \frac{\overline{t} \overline{x^2} - \overline{t} \overline{x} \overline{x} - \overline{x} \overline{xt} + \overline{x} \overline{x} \overline{t}}{\overline{x^2} - \overline{x} \overline{x}} \\ &= \frac{\overline{t} \overline{x^2} - \overline{x} \overline{xt}}{\overline{x^2} - \overline{x} \overline{x}} \end{aligned}$$

它即为式 (1-17) 中需要的第一行。

1.3.2 数值的例子

为了有助于不熟悉向量和矩阵的读者理解, 我们在此重复前面章节中给出的线性回归的例子。矩阵中的数据为:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 5 \end{bmatrix}, \mathbf{t} = \begin{bmatrix} 4.8 \\ 11.3 \\ 17.2 \end{bmatrix}$$

检查式 (1-16), 我们看到需要计算的第一个量是 $\mathbf{X}^T \mathbf{X}$:

$$\mathbf{X}^T \mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 5 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 5 \end{bmatrix} = \begin{bmatrix} 3 & 9 \\ 9 & 35 \end{bmatrix}$$

用上面的公式计算其逆矩阵为

$$(\mathbf{X}^T \mathbf{X})^{-1} = \frac{1}{24} \begin{bmatrix} 35 & -9 \\ -9 & 3 \end{bmatrix}$$

乘以 \mathbf{X}^T ,

$$(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T = \frac{1}{24} \begin{bmatrix} 35 & -9 \\ -9 & 3 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 5 \end{bmatrix} = \frac{1}{24} \begin{bmatrix} 26 & 8 & -10 \\ -6 & 0 & 6 \end{bmatrix}$$

最后，此矩阵乘以 t ：

$$((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T) t = \frac{1}{24} \begin{bmatrix} 26 & 8 & -10 \\ -6 & 0 & 6 \end{bmatrix} \times \begin{bmatrix} 4.8 \\ 11.3 \\ 17.2 \end{bmatrix} = \begin{bmatrix} 1.8 \\ 3.1 \end{bmatrix}$$

因此，其结果为 $f(x; w_0, w_1) = 1.8 + 3.1x$ ，与前面一样正确。

1.3.3 预测

给定属性 x_{new} 的一个新向量，对于模型 t_{new} 的预测可以计算为：

$$t_{\text{new}} = \hat{\mathbf{w}}^T \mathbf{x}_{\text{new}}$$

1.3.4 小结

在前面的各节中，我们介绍了用向量和矩阵描述的线性模型。结果得到一个非常有用的模型—— $\hat{\mathbf{w}}$ 的表达式并没有对 $\hat{\mathbf{w}}$ 的参数个数（它的长度）做假设。因此可以计算 $\hat{\mathbf{w}}$ ，并对其任意形式的线性模型做预测：

$$t_n = w_1 x_{n1} + w_2 x_{n2} + w_3 x_{n3} + \dots$$

这是一个很有效的工具——许多真实的数据集往往不止一个属性，对其中大多数属性来说，该线性模型是适用的。我们也了解到该模型的预测是非常精确的，但不总是合理的。在后续的章节中将考虑如何克服这个缺点。

组成 \mathbf{x}_n 的属性衡量不同的特性（例如，获胜的次数和个人的最好成绩）。或者说，可能是把一个函数集应用到奥运会年份这个单独属性的结果： x_n 。它允许扩展线性模型，这是下一节将要讨论的主题。

24

1.4 线性模型的非线性响应

从这章开始，假设应用线性函数对时间和奥运会 100 米短跑时间之间的关系建立模型。在很多实际应用中，这个太受约束。即使对于 100 米数据，它也表明它过于简单化——线性模型预测 3000 年，时间将会是 -3.5 秒！幸运的是，对于更多复杂的模型，可以通过属性转换正确地使用我们之前描述的相同的框架。

到目前为止，可以看到的线性模型

$$f(x; \mathbf{w}) = w_0 + w_1 x$$

是一个关于参数 (\mathbf{w}) 和数据 (x) 的线性关系式（见注解 1.1）。从计算的角度看，参数的线性性可以描述为最小化平方损失函数的解，如式 (1-8) 和式 (1-10) 中。增加列 x_n^2 ，扩展数据矩阵 \mathbf{X} ：

$$\mathbf{x}_n = \begin{bmatrix} 1 \\ x_n \\ x_n^2 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 \end{bmatrix}$$

并且增加一个额外参数 \mathbf{w} ：

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

结果为：

$$f(x; \mathbf{w}) = \mathbf{w}^T \mathbf{x} = w_0 + w_1 x + w_2 x^2$$

由于参数中的模型仍是线性的，所以可以用式 (1-16) 来求取 \mathbf{w} ，但是要拟合的函数在数据中是二次的。图 1-9 给出了使用该方法的例子，它用数据的二次函数来拟合适当的数据集（实线）（MATLAB 脚本：synthquad.m）。它还说明通过尝试获得的函数能够拟合原始线性（在数据中）模型（虚线， $t = w_0 + w_1 x$ ）。很明显，从拟合的结果看，二次模型更适合。

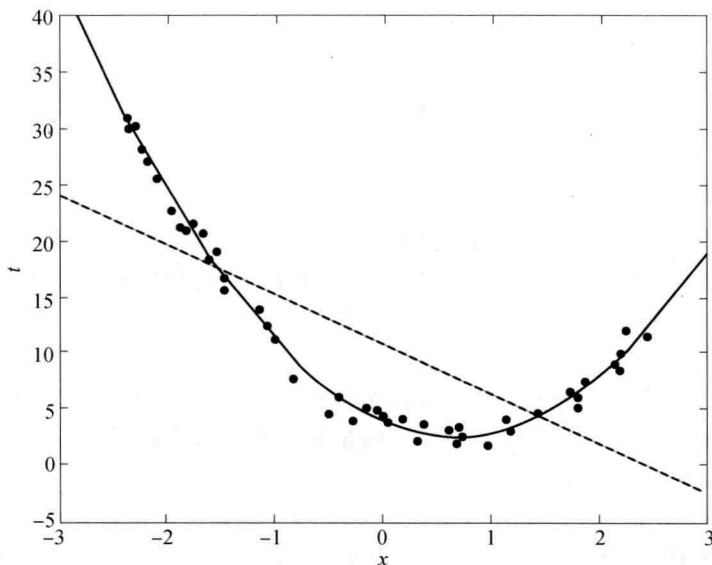


图 1-9 线性和二次模型拟合二次函数生成的数据集

更一般地，可以增加 x 的幂得到任意阶的**多项式函数**。对于一个 K 阶多项式，可以扩展数据矩阵为：

$$\mathbf{X} = \begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & \cdots & x_1^K \\ x_2^0 & x_2^1 & x_2^2 & \cdots & x_2^K \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_N^0 & x_N^1 & x_N^2 & \cdots & x_N^K \end{bmatrix} \quad (1-18)$$

（其中 $x^0=1$ ），函数可以表达为更一般的形式：

$$f(x; \mathbf{w}) = \sum_{k=0}^K w_k x^k$$

图 1-10 给出了前面提到的 100 米短跑数据拟合 8 阶多项式函数的效果（MATLAB 脚本：olymppoly.m）。与图 1-5 和图 1-6 相比，8 阶模型的效果是否比 1 阶的效果更好？为了回答这个问题，需要更精确地了解所谓更好的含义。对模型做预测问题，论证认为产生最好的预测的模型是最好的。关于模型选择的细节问题，可以参见 1.5 节。然而，有两个问题显而易见，值得说明。第一，8 阶多项式比 1 阶多项式（原始模型）更接近观测数据。这反映了有更低的损失函数值： $\mathcal{L}^8 = 0.459$ ， $\mathcal{L}^1 = 1.358$ （其中 \mathcal{L}^k 是 k 阶多项式的损失）。事实上，增加多项式的阶会导致模型更加接近训练数据。第二，预测（实线所显示的）不够合理，尤其在观测数据范围之外。

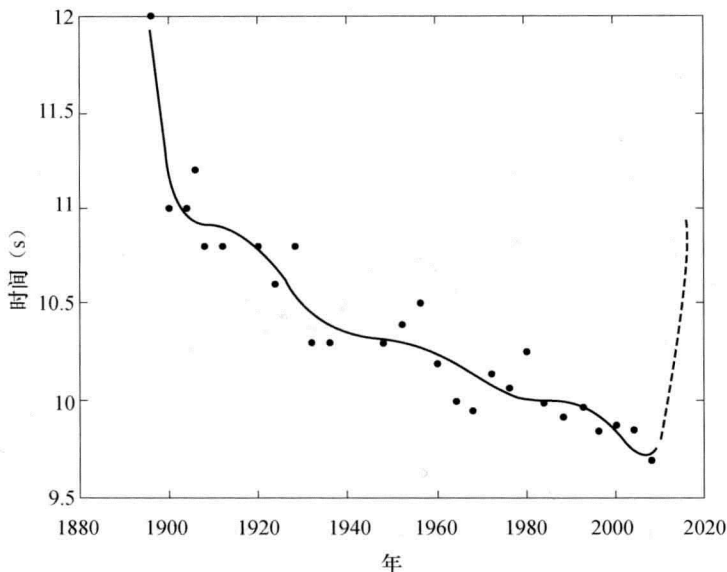


图 1-10 8 阶多项式拟合奥运会 100 米男子短跑数据

不限于多项式函数，我们可以随意定义 x 的任何 K 项函数集 $h_k(x)$ ：

$$\mathbf{X} = \begin{bmatrix} h_1(x_1) & h_2(x_1) & \cdots & h_K(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_K(x_2) \\ \vdots & \vdots & \cdots & \vdots \\ h_1(x_N) & h_2(x_N) & \cdots & h_K(x_N) \end{bmatrix}$$

它适用于任何可用的数据。例如，在 100 米比赛数据中出现周期趋势，合适的函数集可表示为：

$$h_1(x) = 1$$

$$h_2(x) = x$$

$$h_3(x) = \sin\left(\frac{x-a}{b}\right)$$

$$f(x; \mathbf{w}) = w_0 + w_1 x + w_2 \sin\left(\frac{x-a}{b}\right)$$

这个模型有 5 个参数： w_0 、 w_1 、 w_2 、 a 、 b 。不幸的是，只有前 3 个参数可以推导出来。后两个参数 a 、 b 出现在非线性（正弦）函数内。因此，对这些参数求偏导数让其结果等于 0 将得不到由解析法求解的方程组。有很多克服该问题的方法，最简单的就是在合理范围内搜索 a 和 b 的所有值。然而，目前将忽略这个问题，假定已知其值。如果固定 a 和 b 的值，那么可以用前面推导的表达式设置其他的参数（ w_0 、 w_1 、 w_2 ）。假设 a 和 b 是固定的（ $a = 2660$ ， $b = 4.3$ ），图 1-11 给出了使用此模型的最小二乘法拟合。在 $\mathcal{L} = 1.1037$ 的情况下，拟合观测数据比 1 阶多项式好，但不如 8 阶多项式。图 1-11 可以很明显地看出模型的各个分量：常数项（ $w_0 = 36.610$ ）、向下的线性趋势（ $w_1 = -0.013$ ）和导致振荡的非线性正弦曲线项（ $w_2 = -0.133$ ）。注意， w_0 和 w_1 的值与 1 阶多项式模型的那些值非常相似（见图 1-5）——可以对原始线性模型增加一个振荡分量。

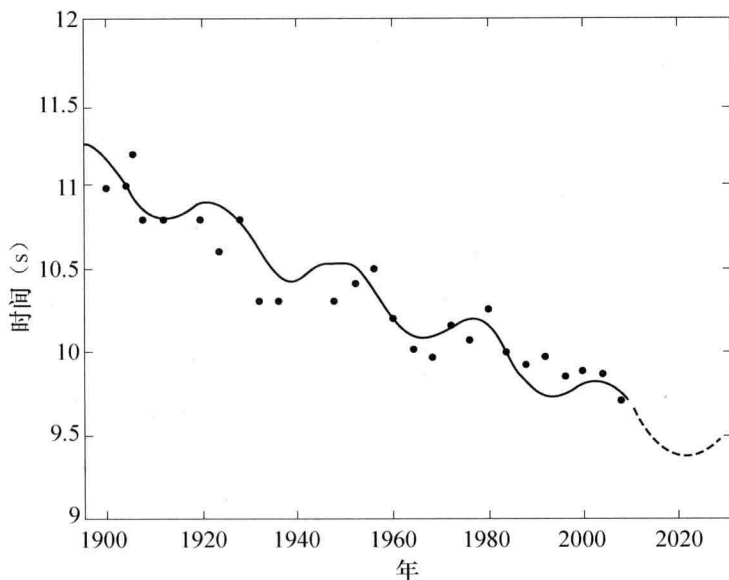


图 1-11 $f(x; \mathbf{w}) = w_0 + w_1 x + w_2 \sin\left(\frac{x-a}{b}\right)$ 的最小二乘法拟合 100 米短跑数据 ($a=2600$, $b=4.3$)

1.5 泛化与过拟合

1.4 节提出了 1 阶与 8 阶多项式哪个更好的问题。假定原来建立这些模型的目的是做预测，那么不难理解最好的模型就是可以使预测最精确的那个，即可以泛化训练样本以外数据的模型（例如，到 2008 年的奥运会数据）。理想情况下，我们更喜欢选择在不可见数据上性能最好的模型（即最小化损失），但是由于问题本身的原因，数据无法得到。

图 1-10 表明，可应用训练数据上的损失选择用于预测的模型。曲线显示训练数据上 8 阶多项式拟合男子 100 米数据的损失比 1 阶多项式更低。而 8 阶多项式对于未来奥运会的预测非常糟糕。基于 8 阶多项式的模型过于关注训练数据（过拟合），因此不能很好地泛化新数据。由于模型越来越复杂，所以也越来越逼近可观测数据。不幸的是，当超过某点，预测的质量就会迅速退化。为了克服过拟合，能够很好地泛化，确定最优模型的复杂度将会非常有挑战性。这个折中问题经常被认为是偏差-方差平衡，将在 2.8 节中简单地介绍。

28

1.5.1 验证数据

克服过拟合问题的一般方法是使用第二个数据集，即**验证集**。用验证集来验证模型的预测性能。验证数据可以单独提供或者从原始训练集中拿出一部分。例如，在 100 米数据中，可以从训练集中拿出 1980 年以后的所有奥运会数据作为验证集。为了进行模型选择，可以在缩小的训练集上训练每一个模型，然后计算它们在验证集上的损失。图 1-12a、b 依次给出了训练和 (log) 验证损失的曲线。训练损失随着多项式阶（**模型复杂度**）的增加单调递减。而验证损失随着多项式阶的增加而快速增长，这表明 1 阶多项式有最好的**泛化能力**，能够产生最可靠的预测。很容易测试这个假设。在图 1-13 中，可以看到数据集（已标记的训练集和验证集）与 1 阶、4 阶和 8 阶多项式函数（MATLAB 脚本：olympval.m）。1979 年已经执行了这个任务，很明显 1 阶模型的确能够给出最好的预测。

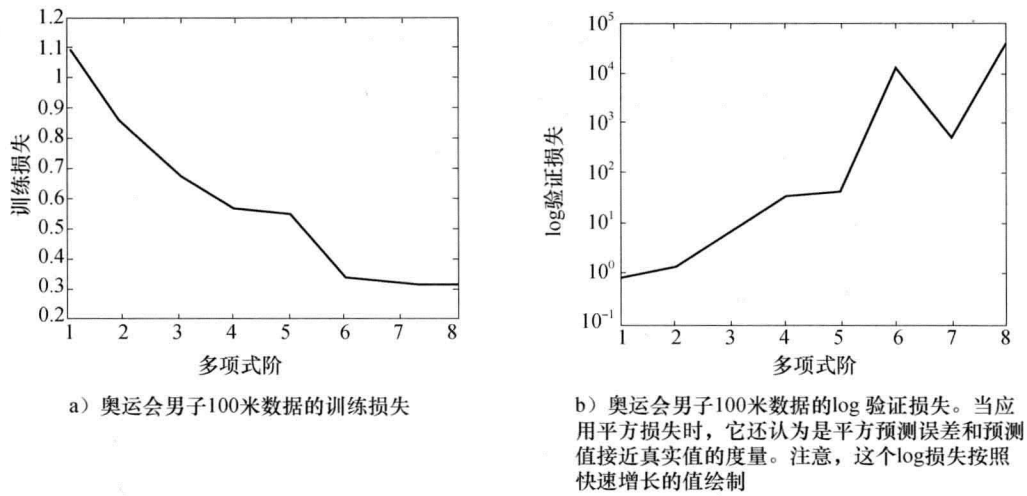


图 1-12 奥运会男子 100 米数据的训练和验证损失

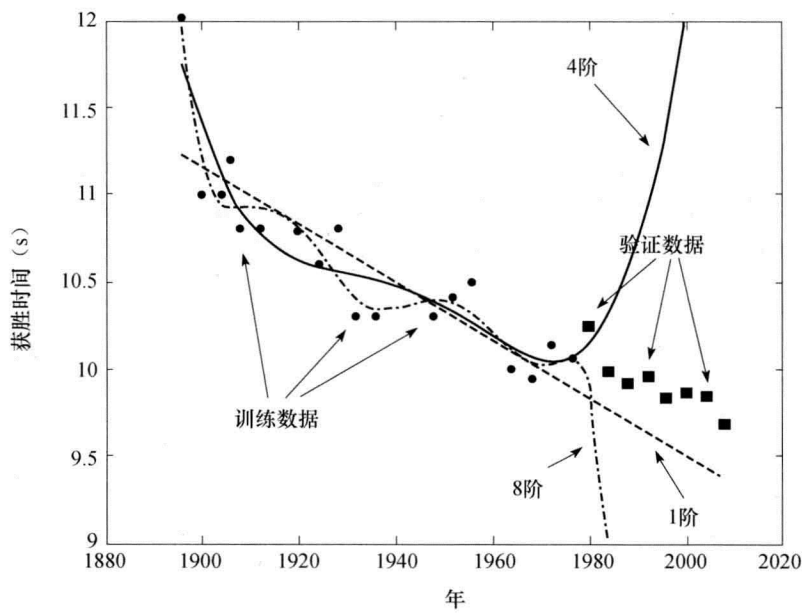


图 1-13 奥运会男子 100 米数据 1 阶、4 阶和 8 阶多项式的泛化能力

1.5.2 交叉验证

从验证集计算的损失对于验证集数据的选择敏感。如果验证集很小，那么更加困难。交叉验证是一种有效使用现有数据集的方法。

如图 1-14 所示， K 折交叉验证把数据集分成大小相等的 K 份（或者尽可能相等）。每块轮流作为验证集，其他 $K-1$ 块作为训练集。结果 K 个损失值的平均值作为最后的损失值。 K 折交叉验证的一个极端情况是，当 $K=N$ ，即 K 恰好等于数据集中的可观测数据的数量时，每个观测数据依次拿出用作测试其他 $N-1$ 个对象训练得到的模型。交叉验证的这种特殊形式称为留一交叉验证（Leave-One-Out Cross Validation, LOOCV），对于 LOOCV 的均方验证为：

$$\mathcal{L}^{CV} = \frac{1}{N} \sum_{n=1}^N (t_n - \hat{\mathbf{w}}_{-n}^T \mathbf{x}_n)^2 \tag{1-19}$$

其中 $\hat{\mathbf{w}}_{-n}$ 是除去第 n 个训练样例的参数估计值。

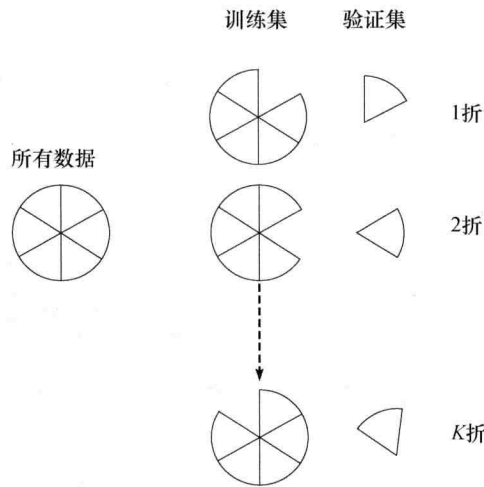


图 1-14 交叉验证。数据集如左边的饼图所示。在每一个 K 折，数据点的一个集合从训练集中移出，用于验证或测试模型

图 1-15 给出了奥运会男子 100 米数据的平均 LOOCV 误差。该曲线表明 3 阶多项式最好，这与最后少量数据点作为验证集的结果不一致。这样的分歧并不少见——**模型选择**就是一个很困难的问题。然而，这两个方法都认为模型不应该是 6 阶或者更高阶。

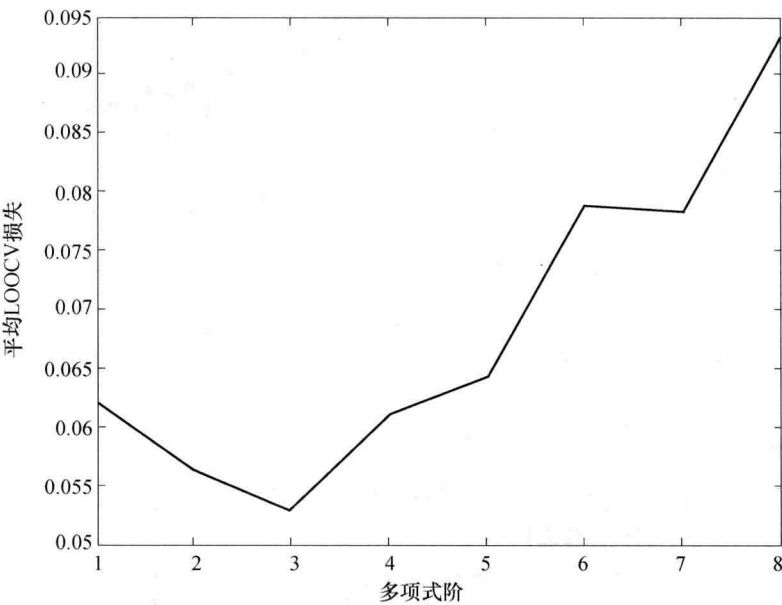


图 1-15 平均 LOOCV 损失作为递阶多项式与奥运会男子 100 米数据拟合

实际数据集上的模型选择问题的一个缺点是并不知道真实模型是什么样的，因此不知道选择技术是否有效。但可以通过生成一个合成数据集克服这个困难。用带噪声的 3 阶多项式函数生成的 50 个输入目标对学习递增（从 1~7）多项式函数。理想情况下，希望看到 3 阶

多项式函数的最小验证损失。再从真实函数/原函数中生成 1000 个输入目标对，作为独立的测试集计算额外的独立损失。与 LOOCV 损失比较，这个大数据集能够给出一个很好的近似正确的期望损失。

图 1-16 给出了上面的结果 (MATLAB 脚本: cv_demo.m)。正如我们看到的，训练损失随着阶的增加而减小。当阶增加到 3 阶 (包括 3 阶) 之前 LOOCV 损失和测试损失会减小，然后再随着阶的增加而增加。这些验证方法都可以预测正确模型的阶。不幸的是，我们很难从训练集外获得 1000 个独立的点，在很大程度上这将依赖于交叉验证方案，通常是 LOOCV。

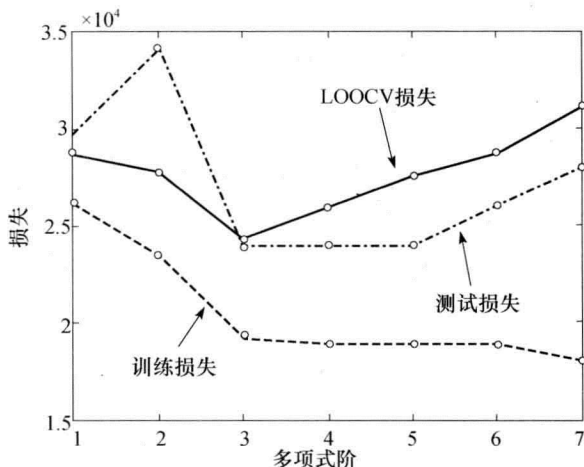


图 1-16 在 50 个样本大小下，从带有噪声的立方函数获得的训练、测试和留一损失曲线作为训练和 LOOCV 估计。使用 1000 个独立样本计算测试误差

1.5.3 K 折交叉验证的计算缩放

留一 (LOO) 交叉验证似乎是估计训练数据集期望损失的一种好方法，它可以查看和评估各种可选择的模型。然而，考虑 LOOCV 的实现。需要训练模型 N 次，这比只在所有数据上训练一次多耗费大约 N 倍的时间 (这样说并不完全准确，因为训练是在小的数据点上进行的)。对于某些模型，尤其是有很多数据的模型，该方法可能并不可行。

缓解这个问题最简单方法就是让 $K \ll N$ 。例如，在 10 折交叉验证中，可以用其中 10% 的数据做验证，其他剩下的 90% 做训练。这样会降低从 $N \sim 10$ 的训练循环数——如果 $N \gg 10$ ，那么将是一个相当大的节省。通常的选择是用 N 折交叉验证，并且应用不同分割的 N 组数据多次反复，对每次重复的验证结果求平均作为最后的结果。

1.6 正则化最小二乘法

在前面的章节中，我们讨论了使用部分训练数据进行预测以确保好的预测性能 (好的泛化)，避免模型的过拟合。本质上，也可以防止模型过于复杂。然而，正则化方法也可以实现这个功能。

定义一个简单模型 $f(x; \mathbf{w}) = \mathbf{w}^T \mathbf{x}$ ，其中 $\mathbf{w} = [0, 0, \dots, 0]^T$ ——该模型总会预测出 0 值。这是最简单的模型。对 \mathbf{w} 元素做任何改变都会增加它们的绝对值，使模型变得更复杂。具体来说，考虑 5 阶多项式模型

$$f(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + w_5 x^5$$

如果让 w 所有元素的初值都为0,那么函数总会预测出0值。假设 w_0 为某个非0值。现在模型就会预测出一个常量(w_0)。让 w_0 为其新值,设置 w_1 为某个值。模型已经变复杂了,同时每个额外参数给定一个非0值,模型就会更加复杂。通常, w 的绝对值总和越大,其模型也就越复杂(注意,是绝对值——不希望正数值消除负数值)。另外,由于绝对值往往在数学上表现得更复杂一些,所以定义模型的复杂度为

$$\sum_i w_i^2$$

或者,表示为向量形式,

$$w^T w$$

由于不希望模型过于复杂,所以让这个值小是有意义的。因此,不仅仅要减小平均平方损失 \mathcal{L} ,更要减小由之前损失和复杂度惩罚项造成的正则化损失 \mathcal{L}' :

$$\mathcal{L}' = \mathcal{L} + \lambda w^T w \quad (1-20)$$

参数 λ 控制数据拟合程度惩罚项(\mathcal{L})和模型复杂程度惩罚项 $w^T w$ 的折中。可以精确地获得与前面方法相同的 w 的最优值。在原来的平方损失(式(1-14))中增加一个正则化项:

$$\mathcal{L}' = \frac{1}{N} w^T X^T X w - \frac{2}{N} w^T X^T t + \frac{1}{N} t^T t + \lambda w^T w$$

求取关于 w 的偏导数

$$\frac{\partial \mathcal{L}'}{\partial w} = \frac{2}{N} X^T X w - \frac{2}{N} X^T t + 2\lambda w$$

让其结果等于0,求 w

$$\frac{2}{N} X^T X w - \frac{2}{N} X^T t + 2\lambda w = 0$$

$$(X^T X + N\lambda I) w = X^T t$$

因此,由正则化最小二乘法得到:

$$\hat{w} = (X^T X + N\lambda I)^{-1} X^T t \quad (1-21)$$

很明显,如果 $\lambda=0$,则和原来的答案一样。可以在合成的例子中增加 λ 的值看其效果。图1-17给出了6个合成数据点。可以看到,如果让 $\lambda=0$,那么5阶多项式函数可以精确地

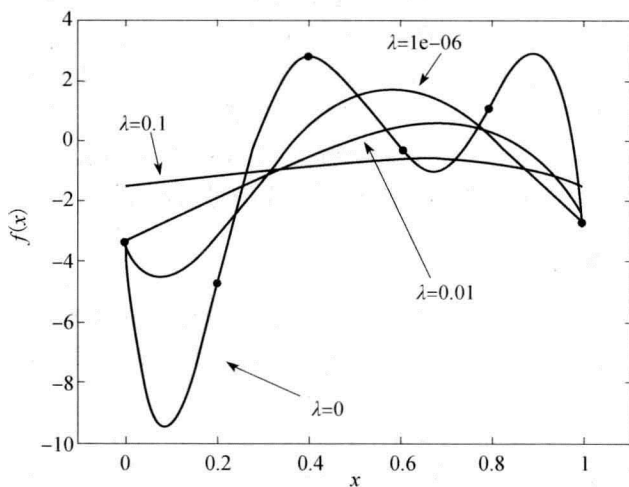


图 1-17 5 阶多项式函数随正则化参数 λ 变化的不同效果

拟合这6个数据点（一般情况下， $N-1$ 阶多项式可以完美地拟合 N 个数据点）。如果开始增加 λ 的值，看看正则化的效果。当 $\lambda=1e-06$ 时，遵循5阶多项式的大体形状但是没有那么多变化，因此结果偏离了数据点。 $\lambda=0.01$ 和 $\lambda=0.1$ 继续这个趋势——函数变得不那么复杂（MATLAB脚本：regls.m）。

选择 λ 值与选择多项式阶时对于过拟合/泛化的折中是一样的。如果值太小，函数就可能太复杂；值太大，又不利于逼近数据。幸运的是，可以准确地使用前面章节中介绍的验证技术确定 λ 的最佳值。特别是，一般采用交叉验证选择能够获得最好预测性能的 λ 值（见练习EX 1.12）。

34

1.7 练习

EX 1.1 根据图1-1估计 w_0 和 w_1 的取值类型。（如高？低？正值？负值？）

EX 1.2 对任意 (x_n, t_n) 对组成的数据集，编写一段Matlab脚本来计算 w_0 和 w_1 的值。

EX 1.3 证明：

$$\mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} = w_0^2 \left(\sum_{n=1}^N x_{n1}^2 \right) + 2w_0 w_1 \left(\sum_{n=1}^N x_{n1} x_{n2} \right) + w_1^2 \left(\sum_{n=1}^N x_{n2}^2 \right)$$

其中，

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \\ \vdots & \vdots \\ x_{N1} & x_{N2} \end{bmatrix}$$

（提示：可先计算 $\mathbf{X}^T \mathbf{X}$ ）。

EX 1.4 使用上一题（EX 1.4）中定义的 \mathbf{w} 和 \mathbf{X} ，通过展开等式两端来证明 $(\mathbf{X}\mathbf{w})^T = \mathbf{w}^T \mathbf{X}^T$ 。

EX 1.5 当向量（或者矩阵）乘以一个标量的时候，仅需对每个向量（或者矩阵）的每个元素分别乘以这个向量。给定 $\mathbf{x}_n = [x_{n1}, x_{n2}]^T$ 、 $\mathbf{t} = [t_1, \dots, t_N]^T$ 、 $\mathbf{w} = [w_0, w_1]^T$ 和

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}$$

证明：

$$\sum_n \mathbf{x}_n t_n = \mathbf{X}^T \mathbf{t}$$

和

$$\sum_n \mathbf{x}_n \mathbf{x}_n^T \mathbf{w} = \mathbf{X}^T \mathbf{X} \mathbf{w}$$

EX 1.6 使用表1-3提供的数据，找出减小平方损失的线性模型。

EX 1.7 根据上一题（EX 1.6）中获得的模型，预测2012年和2016年奥运会女性获胜的时间。

EX 1.8 使用男子和女子100m的模型，找出女子超越男子的奥运会项目。预测的获胜次数是多少？预测结果现实吗？

35

EX 1.9 使用synthdata.mat数据，拟合4阶多项式函数 $f(x; \mathbf{w}) = w_0 + w_1 + w_2 x^2 + w_3 x^3 + w_4 x^4$ 。你发现 w_2 和 w_4 有什么特点？使用10折交叉验证来选择多项式的阶数（从1~4）。

EX 1.10 推导出最优最小二乘法的参数值 $\hat{\mathbf{w}}$ ，对于所有的训练损失：

$$\mathcal{L} = \sum_{n=1}^N (t_n - \mathbf{w}^T \mathbf{x}_n)^2$$

比较该公式与平均损失函数有什么不同？

EX 1.11 下面的公式称为加权平均损失：

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N \alpha_n (t_n - \mathbf{w}^T \mathbf{x}_n)^2$$

其中，每个数据点的作用由相应的参数 α 决定。假设每个 α_n 都是固定的，推导最优最小二乘法的参数 $\hat{\mathbf{w}}$ 。

EX 1.12 使用 K 折交叉验证找到最优的 λ ，对奥运会男子 100 米数据具有最佳的预测结果，

a 1 阶多项式（即标准线性模型）

b 4 阶多项式模型

36

其他阅读材料

- [1] F. Galton. Regression towards mediocrity in hereditary stature. *Anthopological Miscellanea*, 15:246–263, 1886.

“回归”一词由 Francis Galton 首次在遗传学背景下提出。这是 Galton 从 1886 年以来关于回归的最初遗传学论文之一。

- [2] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition, 2009.

这本书中有一章详细地讲解了最小二乘技术，这个技术是进一步开发这个想法的好起点。

- [3] K. B. Petersen and M. S. Pedersen. The matrix cookbook.
<http://www2.imm.dtu.dk/pubdb/p.php?3274>, October 2008.

提供了许多有用的矩阵恒等式的优秀免费资源。特别是，一个提供许多矩阵公式的极好的免费资源。希望对多元高斯密度函数的使用特别有用。

37

线性建模：最大似然方法

第1章介绍了通过定义和最小化损失函数来学习模型参数的方法。在本章的末尾，我们将从不同的起点推导出完全相同的方程来优化模型参数。特别地，我们引入一个随机变量来显式地对数据中的噪声（模型和观测值之间的误差）建模。同时说明在模型中引入噪声项的可观优势。本章的大部分（2.2~2.5节）介绍随机变量和概率的相关内容，已经有相关知识的读者可以跳过本部分。

2.1 误差作为噪声

在图1-5中，我们看到使用线性模型通过最小化损失函数来建模奥运会100m数据的结果。线性模型看上去好像能够捕捉到令人关注的下降趋势，但是它不能完美地解释每一个数据点——因为模型和实际数据之间存在误差。在图2-1中标注了这些误差。

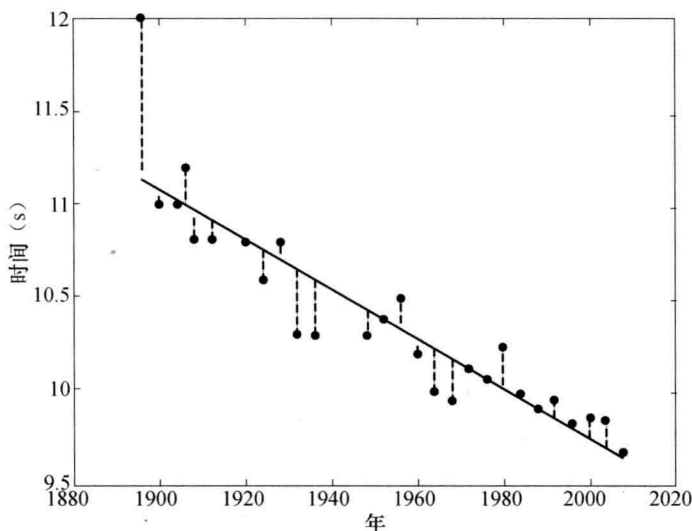


图 2-1 奥运会男子 100 米数据的线性拟合，标注了误差

在构建模型的时候，我们假设年和比赛时间存在线性关系。这个模型看起来能够捕获数据中的总体趋势，同时忽略模型和观测数据之间出现的较大偏差。从建模的观点来看，很难忽略这些误差。如果我们知道如何表示这些误差，那么我们应该努力将它们构建在模型中。

本章将看到显式地对这些误差建模的好处。特别地，这将允许我们在估计模型参数 w 时表达不确定性的级别——如果稍稍改变 w ，这还是一个好的模型吗？这反过来允许我们表达预测中不确定性的程度——“我们相信获胜时间将在 a 、 b 之间”而不是“我们认为获胜时间一定是 c ”。

2.1.1 产生式的考虑

产生这个特定数据集的过程很复杂——我们甚至不能开始构造一个近乎完美的模型来表

示一个短跑运动员以及影响其准备和表现的事件，更不用说多个运动员和所有其他因素了。然而，将建模问题当做产生式模型仍然是有用的：我们能建立一个模型，使得这个模型产生与现有数据相似的数据集吗？虽然我们承认实际数据不是这样产生的这个事实，但我们将会发现这是一个有用的策略。

我们怎样着手从现在的模型生成数据呢？对于等式 $f(x; \mathbf{w}) = \mathbf{w}^T \mathbf{x}$ ，如果代入第1章得到的 \mathbf{w} 值，那么这个等式就能够针对每一个特殊的年份产生一个获胜时间。图2-2给出了这种方法产生的1920—2000年的获胜时间。这与图2-1的数据看上去并不是很像。为了让这组数据更贴近现实，需要增加一些误差。检查图2-1，我们发现这些误差有两个重要特征。

- 1) 每年的误差都不同。有些是正的有些是负的，并且大小不同。
- 2) 误差的大小或方向与年份之间并没有明显的关系。误差不像是奥运会年份 x 的函数。

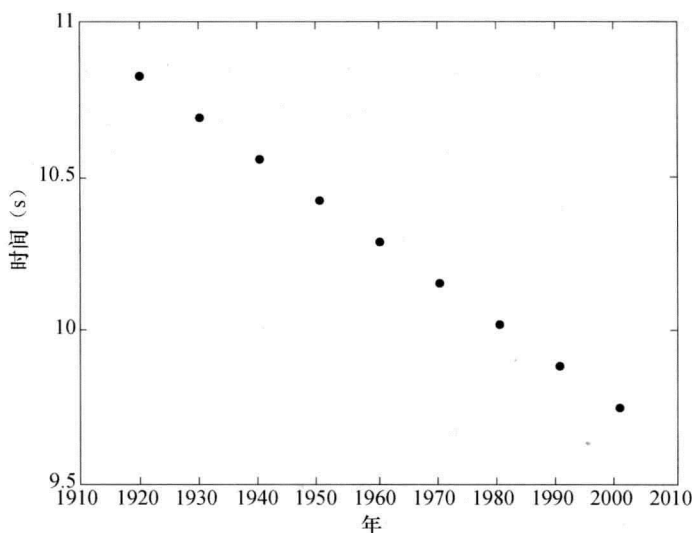


图 2-2 使用线性模型生成的数据集

如果有方法产生或正或负的随机大小的时间（秒级），并且这个时间与图2-1的误差大致相等，那么我们就能够针对每一个数据点生成这样一个我们想要的时间并且可以将它加到 $\mathbf{w}^T \mathbf{x}$ 上。将我们需要的可变性引入模型的工具来源于统计学。下一节我们将介绍随机变量和使用随机变量的几种方法。熟悉这些内容的读者可以跳至2.6节。

2.2 随机变量和概率

我们建立的任何模型都是产生观测数据的真实系统的简化。这导致了模型和现实的差异。本节介绍的工具将帮助我们建模和理解这种差异。因为我们必须从基础开始，所以开始的时候它可能与将误差加入100米数据和表达预测不确定性这个特定问题无关。但是随着进一步深入，这种联系将逐渐清晰。

2.2.1 随机变量

方程

$$y = 5x - 2$$

有 x 和 y 两个变量。如果已知其中一个（例如 $y = 8$ ），那么就能够求解出另一个（ $x = 2$ ）。

随机变量与此很不相同。随机变量允许我们对随机事件指派数值。例如，想要对抛硬币的结果建模。开始时设置一个变量 X ，如果正面朝上赋给 X 值 1，否则为 0。 X 是一个随机变量——“变量”部分描述了 X 可以取不同值的事实（这里是 0、1），而所谓的“随机”是指在抛掷硬币发生之前我们不知道 X 取什么值。我们不能用与标准变量函数相类似的方式来表达这个结果（例如， $y = 5x - 2$ ）。按照惯例，我们用大写字母来表示随机变量，用小写字母来表示这个随机变量可以取的值。

随机变量有两种类型，并且这两种随机变量处理起来稍有不同。离散随机变量是最容易理解的概念，它们用来表示随机事件，对于这些随机事件，我们可以系统地列出随机事件所有可能的结果。例如，离散随机变量可以用来描述投硬币（可能的结果为 0、1），或者掷骰子（可能的结果为 1~6）。所以可能结果的集合叫做样本空间。

能够有组织地按顺序写出所有的事件似乎应该适用于任何的事情。实际上，有许多可能发生的事件并非如此。以奥运会男子 100m 为例，假设获胜时间为 9~10 秒。我们尝试系统地写下所有的可能：

9, 9.1, 9.2, ...

在某一点上，我们意识到我们错过了一些（例如，所有 9~9.1 之间的可能），因此我们重新开始：

9, 9.01, 9.02, ..., 9.1, ...

但是 9~9.01 的值怎么办？第三次尝试：

9, 9.001, 9.002, 9.003, ..., 9.01, ...

这个事件所有可能的结果不能够系统地写出来（每次写下两个值之后，两个值之间都有一些丢失的值）。对于这种事件，我们使用连续随机变量。

表 2-1 给出了我们想要使用随机变量建模的事件或者量，有离散的也有连续的。下面我们通过离散随机变量介绍几个重要概念，之后再将这些概念扩展到连续随机变量情形。

表 2-1 可以使用随机变量建模的事件

过程	离散或连续
抛硬币	离散
投骰子	离散
100 米比赛的结果	连续
计算机网络中的失效节点	离散
诉讼案件的结果	离散
人的身高	连续
卵石的质量	连续
足球比赛的得分	离散
100 米线性回归模型中的误差	见练习 EX 2.1

2.2.2 概率和概率分布

设 Y 是表示抛硬币的随机变量。正面，则 $Y = 1$ ；反面，则 $Y = 0$ 。要对这个事件建模，需要能够量化每一个结果的可能性。对于离散随机变量，我们定义不同结果的概率来达到这个目的。考虑一个特定结果概率的一个直观方法是假设它表示该结果出现的次数与事件重复次数的比例。如果抛公平（即两面朝上可能性一致）硬币 1000 次，那么我们期望看到正面向上的概率大约占一半（剩下的反面朝上）。将正面朝上的概率（ $P(Y = 1)$ ）定义为一半或者 0.5 看起来是合理的。如果硬币不是正面朝上，那么它将反面朝上（在我们的样本空间中只有这两种可能），因此反面朝上的概率是 1 减去正面朝上的比例。因此， $P(Y = 0) = 1 - P(Y = 1) = 0.5$ 。

多次重复实验时，以一个特定结果发生次数的比例作为概率不是我们能想到的定义概率的唯一方法。特别是对于某些只能出现一次的事件，它并不总是最自然的类比。相对于我们的需要它已经足够了，但是鼓励读者进一步研究这个领域。

从我们对比例的简单讨论中，能够得到掌握概率的两条重要规则：

- 概率必须大于或等于 0（比例不能为负数）同时小于或等于 1。
- 所有可能单个结果的概率之和必须等于 1。

例如，抛硬币： $P(Y = 1) + P(Y = 0) = 1$

掷骰子： $P(Y = 1) + P(Y = 2) + \cdots + P(Y = 6) = 1$

这些陈述等同于数学表达式：

$$0 \leq P(Y = y) \leq 1 \quad (2-1)$$

$$\sum_y P(Y = y) = 1 \quad (2-2)$$

依据惯例，小写字母 y 表示随机变量 Y 可以取的所有可能值。注意，我们经常需要书写对随机变量所有取值求和，为了保持符号简明，用 \sum_y 表示对随机变量 Y 所有可能的取值求和。

$P(Y = y)$ 是一个标量——随机变量 Y 取值为 y 的概率。这个符号有时候是不方便的，因此我们有时候使用下面的简写：

$$P(Y = y) = P(y)$$

所有可能取值的集合（所有的 y ）和它们的概率（ $P(y)$ ）称为概率分布。它表明总概率（1）如何在所有可能的结果上分配。

通常，我们使用式（2-1）和式（2-2）基于某些基本假设来定义概率。例如，抛硬币例子中，我们假设正、反两个结果是等可能的： $P(Y = 1) = P(Y = 0) = r$ 。将这个假设代入式（2-2），并且已知 r 属于 $0 \sim 1$ ，我们可以使用基本代数计算出 r 的值（见练习 EX 2.2）。

$$P(Y = 0) + P(Y = 1) = 1$$

$$2r = 1$$

$$r = \frac{1}{2}$$

43

2.2.3 概率的加法

设 Y 是一个随机变量，用来对掷公平骰子的结果建模。如果假设骰子是公平的，即所有结果是等可能的，通过上一节我们知道计算每一个结果（1、2、3、4、5、6）的概率需要足够的知识。掷一次骰子且点数是4，再次投掷骰子，点数小于4的概率是多少？或者我们玩一个打赌的游戏，想要知道是否出现奇数。小于4的所有点数是1、2、3，这表明我们能够计算骰子出现1、2、3点的概率。如果骰子已经投掷了许多次，那么我们能够计算某一点数出现的比例。出现1、2、3点数的比例等于出现1点的比例加上出现2点的比例以及出现3点的比例。这引导我们得出概率的加法定律：

$$P(Y < 4) = P(Y = 1) + P(Y = 2) + P(Y = 3)$$

无论我们感兴趣的结果的顺序如何，都得到了完全一致的答案。例如，投掷出1点或者6点的概率应该是 $P(Y = 1) + P(Y = 6)$ 。这并不仅仅限制在特定的结果上。例如，掷骰子出现不是4点的概率可以这样计算：

$$P(Y \neq 4) = P(Y < 4) + P(Y > 4)$$

$$= P(Y = 1) + P(Y = 2) + P(Y = 3) + P(Y = 5) + P(Y = 6)$$

顺便说一下，值得记住的是一般有多个方法计算概率。在这个例子中，事实上使用式（2-2）计算更简单：

$$P(Y \neq 4) + P(Y = 4) = 1$$

$$P(Y \neq 4) = 1 - P(Y = 4)$$

2.2.4 条件概率

一个事件常常会影响另一个事件的结果。例如，抛一个硬币然后我告诉你结果（你看不

到硬币)。这里有两个事件：第一个，抛硬币；第二个，我告诉你抛硬币的结果。假设这两个事件用两个随机变量 X 和 Y 表示。如果正面朝上， X 为 1；反面朝上， X 为 0。如果我告诉你正面 Y 为 1；否则，为 0。除非我行为怪异，否则， Y 的结果将依赖于 X 的结果。使用条件概率表达已知 X 取特定值的时候 Y 也取特定值的概率。

$$P(Y = y | X = x) \quad (2-3)$$

读作当 X 取值为 x 时， Y 取值为 y 的概率。与非条件概率一样，我们使用如下的简写：

$$P(Y = y | X = x) = P(y | x)$$

44

在我们的例子中，如果假设我总是说真话，那么硬币是正面我告诉你硬币是正面的概率是 1（总是发生）：

$$P(Y = 1 | X = 1) = 1$$

对于反面也是一样：

$$P(Y = 0 | X = 0) = 1$$

使用式 (2-2) 和上述的概率，我们可以推导出 $P(Y = 0 | X = 1)$ 和 $P(Y = 1 | X = 0)$ ：

$$P(Y = 0 | X = 1) + P(Y = 1 | X = 1) = 1$$

$$P(Y = 0 | X = 1) = 1 - P(Y = 1 | X = 1) = 0$$

$$P(Y = 1 | X = 0) + P(Y = 0 | X = 0) = 1$$

$$P(Y = 1 | X = 0) = 1 - P(Y = 0 | X = 0) = 0$$

如果我不诚实，那么事情将会变得更有趣。假设硬币反面朝上我总是说真话，但是如果硬币是正面我说真话（即正面）次数的比例是 0.8。这意味着，如果硬币是正面我说正面的概率是 0.8，说反面的概率是 0.2。在这个假设下，所有的条件概率如下：

$$P(Y = 1 | X = 1) = 0.8$$

$$P(Y = 0 | X = 1) = 0.2$$

$$P(Y = 1 | X = 0) = 0$$

$$P(Y = 0 | X = 0) = 1$$

与非条件概率一样，概率必须满足式 (2-2)，即 $\sum_y P(Y = y | X = x) = 1$ 。检查刚计算的值：

$$\sum_y P(Y = y | X = 1) = P(Y = 1 | X = 1) + P(Y = 0 | X = 1) = 0.8 + 0.2 = 1$$

$$\sum_y P(Y = y | X = 0) = P(Y = 1 | X = 0) + P(Y = 0 | X = 0) = 0 + 1 = 1$$

有条件概率并假设 $P(X = 1) = P(X = 0) = 0.5$ （即硬币是公平的），我们可能会问：“硬币是正面，我说正面的概率是多少？”这与 $P(Y = 1 | X = 1)$ 不同：这个条件概率假设 $X=1$ 已经发生，唯一不确定的是剩下的 Y 将是什么结果。然而我的问题关注这两个事件（ $X=1$ 和 $Y=1$ ）。如果都没有发生，那么它们同时取得特定结果的概率是多少？我们可能需要评估其他感兴趣的量是 $P(Y = 1)$ 和 $P(Y = 0)$ ，即我说硬币是正面或者反面的概率。为此，需要多元概率和多项分布的知识。

2.2.5 联合概率

已知 2 个（或更多）随机变量，我们可能想知道它们每个取得某一特定值的概率。继续讨论之前抛硬币的例子。我们可能想要知道硬币是正面同时我说正面或者硬币是反面同时我说正面的概率。这些是联合概率，定义为：

$$P(Y = y, X = x) \quad (2-4)$$

45

(或者表示成函数形式 $p(y, x)$)。我们如何处理联合分布依赖于这些随机变量是否相关。在我们的例子中, Y (我说的结果) 依赖于 X (抛硬币的结果)。情况就是这样, 即使我不那么诚实, 抛硬币的结果也决定我说什么。如果两个变量没有相关性 (例如, 两个随机变量表示不同的抛硬币事件, 一个结果不可能影响另一个结果), 那么联合概率可以通过两个单独概率相乘来计算:

$$P(Y = y, X = x) = P(Y = y) \times P(X = x)$$

Y 取值 y 并且 X 取值 x 的概率等于 Y 取值 y 的概率乘以 X 取值 x 的概率。更一般地 (为了方便, 我们使用函数形式 $p(y_1, \dots, y_J)$ 而不是 $P(Y_1 = y_1, \dots, Y_J = y_J)$), 对于 J 个随机变量, 有

$$P(y_1, y_2, \dots, y_J) = P(y_1) \times P(y_2) \times \dots \times P(y_J) = \prod_{j=1}^J P(y_j) \quad (2-5)$$

如果这些事件是相互依赖的, 那么我们就不能以这种方式分解联合概率。然而, 如果能建立条件概率分布, 我们就可以使用如下定义来分解联合概率:

$$P(Y = y, X = x) = P(Y = y | X = x) \times P(X = x) \quad (2-6)$$

或者

$$P(Y = y, X = x) = P(X = x | Y = y) \times P(Y = y) \quad (2-7)$$

所以, 硬币正面朝上并且我说正面的概率是

$$P(Y = 1, X = 1) = P(Y = 1 | X = 1) \times P(X = 1) = 0.8 \times 0.5 = 0.4$$

或者, 换句话说, 如果重复多次实验, 那么硬币朝上并且我说朝上的比例是 0.4。我偶尔撒谎的事实使得硬币是正面你也听到正面的概率从 0.5 (我诚实的情况下) 下降至 0.4。

X 和 Y 有 4 种可能的组合, 因此有 4 种可能的结果。式 (2-2) 表明: 如果将这 4 种情况的概率相加, 和应该是 1。

$$\sum_{x,y} P(X = x, Y = y) = 1 \quad (2-8)$$

(注意: $\sum_{x,y}$ 对应着对 x, y 所有 4 种可能组合情况求和)。我们可以使用式 (2-6) 计算所有情况来检测式 (2-8)。我们已经知道 $P(X = 1, Y = 1) = 0.4$ 。其余的是:

$$P(Y = 0, X = 1) = P(Y = 0 | X = 1) P(X = 1) = 0.2 \times 0.5 = 0.1$$

$$P(Y = 1, X = 0) = P(Y = 1 | X = 0) P(X = 0) = 0 \times 0.5 = 0$$

$$P(Y = 0, X = 0) = P(Y = 0 | X = 0) P(X = 0) = 1 \times 0.5 = 0.5$$

根据需要将这些加在一起, 即 $0.4 + 0.1 + 0 + 0.5 = 1$ 。

在继续下面的内容之前, 我们快速考虑这三个值。第一个值 (0.1) 给出了硬币是正面我说是反面的概率。这比我诚实情况下对应的概率有所增加 (我总是说真话时的概率是 0), 因为硬币是正面时我偶尔撒谎。第二个值 (0) 给出了硬币实际是反面而我说是正面的概率。这个值是 0, 因为硬币是反面的时候我从不撒谎。第三个值给出了硬币是反面而我也说是反面的概率。它是 0.5, 因为硬币有一半的次数是反面, 而反面的时候我总是讲真话。

2.2.6 边缘化

如果记录我说正面或者反面次数的比例, 实际上是计算 $P(Y = 1)$ 和 $P(Y = 0)$ 。这些表达式没有包含 X ——它们仅仅涉及我说什么这个事件。 $P(Y = y)$ 可以通过从联合概率 $P(Y = y, X = x)$ 中边缘化 X 得到。这通过对联合概率在 X 的所有可能取值上求和得到:

$$P(Y = y) = \sum_x P(Y = y, X = x) \quad (2-9)$$

在我们硬币的例子中， X 可以取两个值（0、1）中的一个，所以求和就变成

$$P(Y = y) = P(Y = y, X = 0) + P(Y = y, X = 1)$$

一般而言，对于 J 个随机变量的联合概率，为了获得它们中一个的边缘分布 $P(Y_i = y_i)$ ，可以通过如下的公式：

$$P(Y_j = y_j) = P(y_j) = \sum_{y_1, \dots, y_{j-1}, y_{j+1}, \dots, y_J} P(y_1, \dots, y_J) \quad (2-10)$$

表达式中的求和符号看起来有些怪异。它表示对剩余 $J-1$ 个变量（缺少 y_i ）的所有可能情况求和。例如，如果 $J=3$ 并且每个变量只能取值 0、1，那么为了计算 $P(Y_1 = y_1) = p(y_1)$ 就需要对 y_2 和 y_3 的四种不同组合求和：

$$\begin{array}{cc} y_2 & y_3 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{array}$$

47

如果 $J=4$ ，那么这个数就增加到 8：

$$\begin{array}{ccc} y_2 & y_3 & y_4 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{array}$$

一般而言，对于二元变量，组合的数量是 2^{J-1} ，它随着 J 呈指数增长。如果随机变量有 2 个以上的结果，那么情况更坏（例如，对于骰子 6^{J-1} ）。在某些机器学习的概率领域，边缘化非常重要，也具有挑战性，令人振奋的近似方法将在第 4 章看到。

回到硬币的例子， $P(Y = 1)$ 是

$$\begin{aligned} P(Y = 1) &= \sum_x P(Y = 1, X = x) \\ &= P(Y = 1, X = 0) + P(Y = 1, X = 1) \\ &= 0 + 0.4 = 0.4 \end{aligned}$$

并且 $P(Y = 0)$ 是

$$\begin{aligned} P(Y = 0) &= \sum_x P(Y = 0, X = x) \\ &= P(Y = 0, X = 0) + P(Y = 0, X = 1) \\ &= 0.5 + 0.1 = 0.6 \end{aligned}$$

我们也可以使用 $P(Y=1)$ 的值和式 (2-2) 计算 $P(Y=0)$ 的值。这些概率表明，我说正面和反面的次数占总次数的比例。这不同于抛硬币得到正面或者反面的次数占总次数的比例 ($P(X=1) = P(X=0) = 0.5$)。这一矛盾是由于在结果告知过程中的不确定性引起的，

48 本章第二个事件中的人实际上是噪声或者误差的来源。注解 2.1 提供了关于条件概率和边缘化的另一个例子。

注解 2.1 (条件概率和边缘化)：假设我们有一个公平的硬币和两个骰子（其中一个有些与众不同）。我们要使用如下过程生成一个抛硬币事件（ X ）和一个掷骰子事件（ Y ）。首先，抛硬币。如果是正面，则掷 1 号骰子；如果是反面，则掷 2 号骰子。1 号和 2 号骰子不一样，其概率按照如下表格定义：

	1	2	3	4	5	6	
1 号骰子	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$= P(y X = H)$
2 号骰子	$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{12}$	$= P(y X = T)$

所以，掷出 3 点的概率，1 号骰子是 $1/6$ ，2 号骰子是 $1/4$ 。因为如果硬币是正面我们投掷 1 号骰子，反面投掷 2 号骰子，所以有如下的条件概率：

$$P(y|X = H), P(y|X = T)$$

即 Y 的概率分布依赖于 X 的结果。依据式 (2-6) 给出联合分布：

$$p(y, x) = p(y|x)p(x)$$

可以使用上式计算出现 3 点且正面的概率：

$$P(Y = 3, x = H) = P(Y = 3|X = H)P(X = H) = \frac{1}{6} \times \frac{1}{2} = \frac{1}{12}$$

相应地，3 点且反面的概率：

$$P(Y = 3, X = T) = P(Y = 3|X = H)P(X = T) = \frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$$

或者更有趣一点，可以计算 Y 的边缘分布。从我们的定义（式 (2-9)）：

$$P(y) = \sum_x P(y, x) = \sum_x P(y|x)P(x)$$

因此，掷出 3 点的概率是：

$$\begin{aligned} P(Y = 3) &= \sum_x P(Y = 3|x)P(x) \\ &= P(Y = 3|X = H)P(X = H) + P(Y = 3|X = T)P(X = T) \\ &= \frac{1}{6} \times \frac{1}{2} + \frac{1}{4} \times \frac{1}{2} = \frac{5}{24} \end{aligned}$$

2.2.7 贝叶斯规则介绍

49 虽然本章我们不需要这个概念，但是介绍贝叶斯^①规则非常有意义，因为从第 3 章开始将广泛涉及贝叶斯规则。式 (2-6) 的左侧和式 (2-7) 的左侧是相同的，因此右侧也可以画上等号。

$$P(Y = y|X = x)P(X = x) = P(X = x|Y = y)P(Y = y)$$

重新整理，可以得到 $Y = y$ 条件下 X 的概率 ($P(X = x|Y = y)$)，这依赖于 $X = x$ 条件下 Y 的概率。这就是著名的贝叶斯规则：

$$P(X = x|Y = y) = \frac{P(Y = y|X = x)P(X = x)}{P(Y = y)} \quad (2-11)$$

① 以 Reverend Thomas Bayes 的名字命名，他是英国数学家、牧师，他第一个提出条件概率的逆。

在我们的例子中，这是已经知道我说的结果， X 取得特定值的概率（或以 $Y=y$ 为条件， $X=x$ 的概率）。如果想要预测硬币实际的正、反面情况，你很可能对这个公式感兴趣。代入数值可以计算出 $P(X=1|Y=1)$ ：

$$P(X=1|Y=1) = \frac{P(Y=1|X=1)P(X=1)}{P(Y=1)} = \frac{0.8 \times 0.5}{0.4} = 1$$

从上式也可以推导出 $P(X=0|Y=1) = 0$ (式 2-2)。类似地，可以计算 $P(X=0|Y=0)$ ：

$$P(X=0|Y=0) = \frac{P(Y=0|X=0)P(X=0)}{P(Y=0)} = \frac{1 \times 0.5}{0.6} = 0.83$$

从上式可以推导出 $P(X=1|Y=0) = 0.17$ 。

前两个值给出的是我说正面（即 $Y=1$ ）时真实的抛硬币的概率，接下来的两个值是我说不反面（ $Y=0$ ）时抛硬币的真实概率。 $P(X=1|Y=1) = 1$ 说明，我说正面意味着正面是抛硬币的真实结果。 $P(X=0|Y=0) = 0.83$ 表示，如果你听到的是反面，那么硬币出现反面（概率是 0.83）比出现正面（概率是 0.17）的可能性大很多。建模时以这种方法逆转条件非常有用，我们将在第 3 章用到该方法并且进一步讨论它。

2.2.8 期望值

当处理随机变量时，使用一个或者多个值来代表一个分布的特征非常有用。均值就是一个明显的例子——我们期望随机变量采用平均值。平均值是期望值的一个例子。期望值表示随机变量 X 的函数 $f(X)$ 期望得到什么值，定义如下（离散随机变量）：

$$E_{P(x)}\{f(X)\} = \sum_x f(x)P(x) \quad (2-12)$$

例如，如果我们对 X 的期望值（均值）感兴趣，那么 $f(X) = X$ ，并且表达式变为

$$E_{P(x)}\{X\} = \sum_x xP(x)$$

50

对于公平的骰子（ $P(x) = 1/6$ ）， X 的期望值是：

$$E_{P(x)}\{X\} = \sum_x x \frac{1}{6} = \frac{1}{6} + \frac{2}{6} + \dots + \frac{6}{6} = \frac{21}{6} = 3.5$$

从这个例子中我们注意到，期望值不需要是随机变量可能取值中的一个（我们不可能摇到 3.5 点）。

其他函数的期望值可以以完全相同的方式计算。例如， $f(X) = X^2$ 的期望值：

$$E_{P(x)}\{X^2\} = \sum_x x^2 \frac{1}{6} = \frac{1}{6} + \frac{4}{6} + \dots + \frac{36}{6} = \frac{91}{6}$$

随机变量 X 的函数的期望值通常不是函数在 X 的期望值处的取值，明白这一点很重要。数学上， $E_{P(x)}\{f(X)\}$ 不一定等于 $f(E_{P(x)}\{X\})$ 。例如，我们刚才计算了 $E_{P(x)}\{X^2\} = 91/6$ ，它不等于 $(E_{P(x)}\{X\})^2 = (21/6)^2$ 。这两个值在一种情况下相等：当随机变量 X 的函数是一个常数乘以 X 时。在这种情况下，通过简单的代数运算就可以证明这两个值相等：

$$\begin{aligned} f(X) &= aX \\ E_{P(x)}\{f(X)\} &= \sum_x axP(x) \\ &= a \sum_x xP(x) \\ &= aE_{P(x)}\{X\} \\ &= f(E_{P(x)}\{X\}) \end{aligned}$$

另一个重要的情况是，当函数仅仅是一个常数时。这时，因为概率分布对所有可能的结果求

和必须为 1, 所以期望值不再存在。

$$\begin{aligned} f(X) &= a \\ E_{P(x)}\{f(X)\} &= \sum_x a P(x) \\ &= a \sum_x P(x) \\ &= a \end{aligned}$$

最后一种特殊情况是, 函数和的期望值等于每个函数期望值的和, 这将非常有用:

$$\begin{aligned} E_{P(x)}\{f(X) + g(X)\} &= \sum_x (f(x) + g(x)) P(x) \\ &= \sum_x f(x) P(x) + \sum_x g(x) P(x) \\ &= E_{P(x)}\{f(X)\} + E_{P(x)}\{g(X)\} \end{aligned}$$

我们会碰到的两种最常见的期望是均值 (上面定义的 $E_{P(x)}\{X\}$) 和方差。方差度量随机变量如何变化, 定义为实际值与期望值之差平方的期望值:

$$\text{var}\{X\} = E_{P(x)}\{(X - E_{P(x)}\{x\})^2\} \quad (2-13)$$

展开括号里的项, 得出随机变量方差的方便表达式:

$$\begin{aligned} \text{var}\{X\} &= E_{P(x)}\{(X - E_{P(x)}\{x\})^2\} \\ &= E_{P(x)}\{X^2 - 2XE_{P(x)}\{X\} + E_{P(x)}\{x\}^2\} \\ &= E_{P(x)}\{X^2\} - 2E_{P(x)}\{X\}E_{P(x)}\{X\} + E_{P(x)}\{X\}^2 \end{aligned}$$

从第二行到第三行, 我们使用了 $E_{P(x)}\{E_{P(x)}\{f(X)\}\} = E_{P(x)}\{f(X)\}$ 。 $E_{P(x)}\{f(X)\}$ 的值是一个常数 (通过求期望值消除了所有包含 X 的项), 外层的期望是一个常数的期望值, 我们前面已经说明, 常数的期望值等于这个常数。将 $E_{P(x)}\{X\}^2$ 的项合并, 我们得到:

$$\text{var}\{X\} = E_{P(x)}\{X^2\} - E_{P(x)}\{X\}^2 \quad (2-14)$$

一般来说, 方差大的随机变量的值比方差小的随机变量的值离均值更远。

注解 2.2 (向量随机变量): 经常需要定义向量的概率分布。这只不过是定义一个大的联合分布的快捷方式。例如, 随机变量 X_1, X_2, \dots, X_N 的取值可以使用向量 $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ 表示。使用这种快捷方式:

$$p(\mathbf{x}) = p(x_1, x_2, \dots, x_N) = P(X_1 = x_1, X_2 = x_2, \dots, X_N = x_N)$$

虽然 \mathbf{x} 是向量, 但 $p(\mathbf{x})$ 是标量, 就像 $P(X_1 = x_1, X_2 = x_2, \dots, X_N = x_N)$ 一样。

向量随机变量的期望值以完全相同的方式计算 (见注解 2.2)。对一个取向量值 \mathbf{x} 的随机变量 X , 其期望值定义如下:

$$E_{P(x)}\{f(\mathbf{x})\} = \sum_{\mathbf{x}} f(\mathbf{x}) P(\mathbf{x})$$

其中求和符号表示对向量 \mathbf{x} 的所有可能值求和。因此, 均值向量定义如下:

$$E_{P(x)}\{\mathbf{x}\} = \sum_{\mathbf{x}} \mathbf{x} P(\mathbf{x})$$

处理向量的时候, 方差的概念扩展为协方差矩阵, 定义为:

$$\text{cov}\{\mathbf{x}\} = E_{P(x)}\{(\mathbf{x} - E_{P(x)}\{\mathbf{x}\})(\mathbf{x} - E_{P(x)}\{\mathbf{x}\})^T\} \quad (2-15)$$

如果 \mathbf{x} 是一个 D 维的向量, 那么 $\text{cov}(\mathbf{x})$ 是一个 $D \times D$ 维的矩阵。对角线上的元素对应着 \mathbf{x} 每个元素的方差, 对角线外的元素表示不同元素在多大程度上与 \mathbf{x} 一起变化, 即它们对于另

外元素的依赖程度如何。例如，元素 x_d 和 x_e 对应一个大的正值，意味着如果 x_d 增加则 x_e 也增加。如果是一个大的负值，那么意味着它们是相关的，但是向相反方向移动（ x_d 增加则 x_e 减小）。如果是 0 值或者接近 0 值，意味着这两个元素没有关系（它们相互独立）。我们在 2.5.4 节给出关于协方差矩阵和相关密度的例子。与方差一样，协方差表达式可以用如下更方便的形式表示：

$$\begin{aligned}\text{cov}\{\mathbf{x}\} &= \mathbf{E}_{P(\mathbf{x})}\{(\mathbf{x} - \mathbf{E}_{P(\mathbf{x})}\{\mathbf{x}\})(\mathbf{x} - \mathbf{E}_{P(\mathbf{x})}\{\mathbf{x}\})^T\} \\ &= \mathbf{E}_{P(\mathbf{x})}\{\mathbf{x}\mathbf{x}^T - 2\mathbf{x}\mathbf{E}_{P(\mathbf{x})}\{\mathbf{x}\}^T + \mathbf{E}_{P(\mathbf{x})}\{\mathbf{x}\}\mathbf{E}_{P(\mathbf{x})}\{\mathbf{x}\}^T\}\end{aligned}$$

重新整理这个表达式：

$$\text{cov}\{\mathbf{x}\} = \mathbf{E}_{P(\mathbf{x})}\{\mathbf{x}\mathbf{x}^T\} - \mathbf{E}_{P(\mathbf{x})}\{\mathbf{x}\}\mathbf{E}_{P(\mathbf{x})}\{\mathbf{x}\}^T \quad (2-16)$$

2.3 常见的离散分布

目前为止，在我们使用的所有例子中，我们能列出每一个随机变量的所有可能的结果。出于解释的目的这是有用的，但是随着可能结果数量的增加，列出所有可能的结果就不太可能了。在现实中，我们常常要处理许多著名的分布族。每一类分布适用于特定类型的事件，通常，这些分布使用参数来调节它们的特征。本节将介绍常见的并且很可能在机器学习中遇到的离散分布。

2.3.1 伯努利分布

在介绍伯努利 (Bernoulli) 分布之前，我们已经碰到过它多次了。它用于像抛硬币一样的具有两个可能结果的事件。对于随机变量 X ，可以取值为 0 或 1（二元随机变量），将其取值为 1 的概率记为 q ，则伯努利分布可以如下表示：

$$P(X = x) = q^x(1 - q)^{1-x} \quad (2-17)$$

伯努利分布是当 $N=1$ 时二项分布（见 2.3.2 节）的特例。

2.3.2 二项分布

二项分布式扩展了伯努利分布用于定义 N 次试验中观察到的一定数目正面的概率。更一般地，我们可以将它用于任何有两个结果（成功、失败）的事件上。如果有 N 个这类事件，那么二项随机变量 Y 可以取从 0（没有一次成功） $\sim N$ （ N 次都成功）的任意值。观察到一定数目成功事件的概率由下式给出：

53

$$P(Y = y) = P(y) = \binom{N}{y} q^y (1 - q)^{N-y} \quad (2-18)$$

表达式的第二部分看起来与伯努利分布表达式非常像。事实上，如果我们定义 N 个二元结果为 x_1, x_2, \dots, x_N ，那么二项分布表达式的第二部分是 N 个二项概率的乘积：

$$\begin{aligned}\prod_{n=1}^N q^{x_n} (1 - q)^{1-x_n} &= q^{\sum_n x_n} (1 - q)^{N - \sum_n x_n} \\ &= q^y (1 - q)^{N-y}\end{aligned}$$

其中 $y = \sum_n x_n$ 是成功的次数（成功对应 $x_n=1$ ）。二项表达式的第一部分是必需的，因为假设 $y=3$ ，那么有多个 x_1, x_2, \dots, x_N 的可能组合与之对应。 $q^y(1-q)^{N-y}$ 只是表示了多可能中的一个。计算所有可能结果的总和等于乘以这些组合的数目，已知组合数函数 $\binom{N}{y}$ （读作

从 N 中选 y ，详见注解 2.3)。图 2-3 显示了当 $N=50$ 、 $q=0.7$ 时的分布函数。

注解 2.3 (组合)：从 N 中选 y ，记作

$$\binom{N}{y}$$

是从 N 个对象中选出 y 个不同对象的方法数的数学表示法。例如： $\binom{4}{1}$ 等于 4——从 4 个对象中选出一个的方法有 4 种（选择 1 号对象、选择 2 号、选择 3 号、选择 4 号）。 $\binom{4}{2}$ 等于 6——可能的选择是 1 和 2、1 和 3、1 和 4、2 和 3、2 和 4、3 和 4。一般地，我们有

$$\binom{N}{y} = \frac{N!}{y!(N-y)!}$$

其中 $N!$ （读作 N 的阶乘）是

$$\prod_{i=1}^N i = N \times (N-1) \times (N-2) \times \cdots \times 1$$

2.3.3 多项分布

我们前面的两个例子是标量随机变量的分布，现在我们看看这样一个分布——这个分布将概率分配给离散变量的向量。基本思想是完全相同的——分布函数对每一个可能的向量分配概率值，这些概率的和必须为 1。作为向量随机变量的动机，假设你创建一个包含 N 个词的随机文本生成器，并且你想要在这些随机文本上定义一个概率分布。这并不像听起来那样愚蠢——机器学习技术常常用来以这种方式定义文本上的概率分布来分析文本。使用词数的向量是表示文本的一种方式。假设字典里 J 个可能的词，那么这个向量的长度为 J ，第 j 个元素保存字典中第 j 个词在文本中出现的次数。多项分布允许我们定义一个在这样的向量上的分布。设 Y 是一个表示文本的随机变量。一个词数的向量 $y = [y_1, \dots, y_J]^T$ 是随机变量的一个实例，多项分布定义 y 的分布如下：

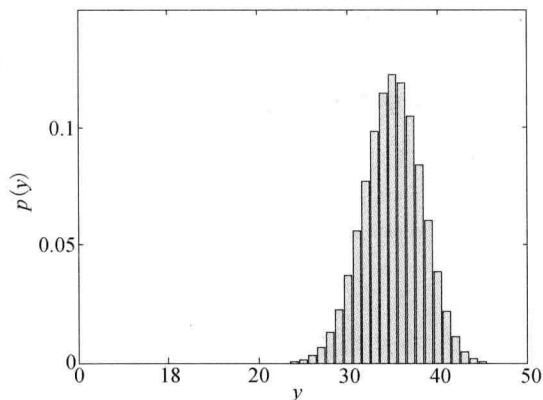


图 2-3 二项随机变量概率分布函数的示例， $N=50$ 、 $q=0.7$ （式 (2-8)）

$$P(Y = y) = P(y) = \frac{N!}{\prod_j y_j!} \prod_j q_j^{y_j} \quad (2-19)$$

其中 q_j 是多项式分布的参数，表示第 j 个词的概率（ $\sum_j q_j = 1$ ）。

2.4 连续型随机变量——概率密度函数

在本章的开始部分我们看到，不能系统地写下连续随机变量所有可能的结果。不幸的是，这也妨碍我们对特定值分配概率。为了解决这个问题，我们使用结果落入某个区间或者间隔的概率。例如，已知一个连续随机变量 X 可以取负无穷大到正无穷大之间的任意值，尝试计算出

$$P(x_1 \leq X \leq x_2)$$

是很有意义的工作，而不是计算

$$P(X = x)$$

当使用连续随机变量时，我们需要概率分布的连续模拟（前面讲过，对离散随机变量，概率分布是一组结果（ x ）和每个结果的概率，表示成 $x, p(x)$ 的函数）。这由概率密度函数（pdf）表示，也记为 $p(x)$ 。为了计算 X 落入某一特定区间的概率，我们计算 $p(x)$ 关于 x 在这个区间上的定积分（见注解 2.4）：

$$P(x_1 \leq X \leq x_2) = \int_{x_1}^{x_2} p(x) dx$$

如果随机变量只是可能在区间 $x_1 \leq X \leq x_2$ 上取值，那么 X 落入这个区间的概率一定是 1。这引导我们得到式（2-2）针对连续随机变量的等价变形：

$$\int_{x_1}^{x_2} p(x) dx = 1 \quad x_1 \leq X \leq x_2 \quad (2-20)$$

式（2-1）也有一个连续随机变量的等价形式：

$$p(x) \geq 0 \quad (2-21)$$

它表明概率密度函数不能为负。值得注意的是，概率密度函数没有上界，因为概率密度函数不是概率，所以对特定的 x 其取值可以（常常是）比 1 大。

56

注解 2.4（定积分）：当对含有常数项的函数微分时，常数项就消失了，例如，

$$\frac{d}{dx}(x^2 + 3) = 2x$$

因此，当我们对一个函数积分时，必须承认这个函数有可能含有常数项

$$\int 2x dx = x^2 + C$$

这就是所谓的不定积分，因为我们不知道 C 的值。

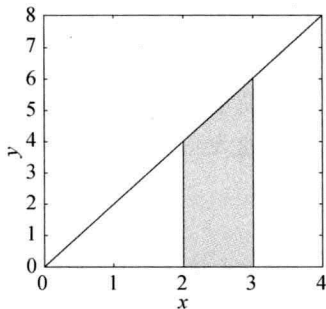
我们常常对使用积分求解曲线下的面积感兴趣。例如，这个例子中我们想要计算 $y=2x$ 在 $x=2$ 和 $x=3$ 之间的面积，如右图所示。此面积依照下式计算：

$$\int_2^3 2x dx = [x^2 + C]_2^3$$

其中 $[\cdot]_a^b$ 表示括号中对象的取值范围为 $x=a$ 到 $x=b$ 。这个例子中，它表示

$$(3^2 + C) - (2^2 + C) = 9 - 4 + C - C = 5$$

这是定积分——消除了常数，并且结果是精确的。



联合概率密度和条件连续概率密度：与离散情况一样，我们可以定义多个连续随机变量的联合概率密度函数。例如， $p(x, y)$ 是连续随机变量 X 和 Y 的联合概率密度， $p(\mathbf{w})$ 是向量 \mathbf{w} 的概率密度函数，向量 \mathbf{w} 的每个元素都是随机变量， $p(\mathbf{w})$ 可以认为是 $p(w_0, w_1, \dots)$ 的联合概率密度函数。尽管我们不能计算 $P(X = x, Y = y)$ ，但是我们可以计算

$$P(x_1 \leq X \leq x_2, y_1 \leq Y \leq y_2) = \int_{x=x_1}^{x_2} \int_{y=y_1}^{y_2} p(x, y) dx dy$$

同样可用于条件分布，即使条件制约是一个确切值（同样，我们假设这个事件已经发生）。例如，我们可能会计算

$$P(x_1 \leq X \leq x_2 | Y = y) = \int_{x=x_1}^{x_2} p(x | Y = y) dx$$

57 我们常常使用简写 $p(x|y)$ 来描述已知 $Y=y$ 时, X 的概率密度函数。

边缘化: 你可能已经在假定连续对随机变量进行边缘化, 我们用积分替代离散随机变量的和。例如, 概率密度函数 $p(y)$ 可以从 $p(y, x)$ 计算出来:

$$p(y) = \int_{x=x_1}^{x_2} p(y, x) dx$$

其中 $x_1 \leq X \leq x_2$ 表示 X 的样本空间。

期望值: 连续随机变量的期望值通过计算随机变量取值范围的积分来完成。

$$E_{p(x)}\{f(x)\} = \int f(x)p(x)dx \quad (2-22)$$

2.28 节中所有推导出的表达式在连续情形下是完全相同的。

在某些实际情况中, 我们可能不能计算这个积分——我们不知道 $p(x)$ 的确切形式, 或者它仅仅无法积分。然而, 如果我们能够对 $p(x)$ 进行采样, 那么可以通过下式近似得到:

$$E_{p(x)}\{f(x)\} \approx \frac{1}{S} \sum_{i=1}^S f(x_s) \quad (2-23)$$

其中 x_s 是从 $p(x)$ 采样得到的 S 个样本。这是蒙特卡罗 (Monte Carlo) 近似的一个例子, 此法在随后章节中我们会进一步讨论。

2.5 常见的连续概率密度函数

与离散情形一样, 我们时常会碰到多类常见的连续概率密度函数。本节中, 我们介绍其中的3种。

2.5.1 均匀密度函数

最简单的连续密度函数是均匀密度函数。均匀密度函数 $p(y) = \mathcal{U}(a, b)$, 在 $a \sim b$ 是常数, 其他为0。

$$p(y) = \begin{cases} r & a \leq y \leq b \\ 0 & \text{其他} \end{cases} \quad (2-24)$$

图2-4是 $a=3$ 、 $b=8$ 时的示例。根据概率密度函数在样本空间上的积分等于1的定义, 对于任意的 a 、 b , 我们能够计算出 r 值。这种情况下,

$$\begin{aligned} P(a \leq Y \leq b) &= 1 = \int_{y=a}^b p(y) dy = \int_{y=a}^b r dy \\ &= [yr]_a^b = rb - ra = r(b-a) \\ r &= \frac{1}{b-a} \end{aligned}$$

这是非常直观的 (r 是全概率1除以区间长度), 因为随机变量必须在 $(b-a)$ 内。我们也可以很容易地定义多维均匀分布随机变量。例如, 如果 $\mathbf{y} = [y_1, y_2]^T$,

$$p(\mathbf{y}) = \begin{cases} r & a \leq y_1 \leq b \text{ 且 } c \leq y_2 \leq d \\ 0 & \text{其他} \end{cases}$$

并且可以以类似的方式计算 r 值,

$$P(a \leq y_1 \leq b, c \leq y_2 \leq d) = 1 = \int_{y_1=a}^b \int_{y_2=c}^d r dy_1 dy_2$$

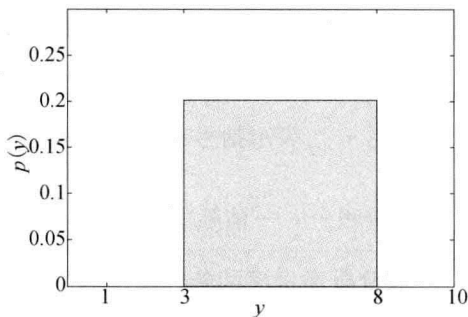


图2-4 均匀概率密度函数的示例

58

$$\begin{aligned}
&= \int_{y_1=a}^b [ry_2]_c^d dy_1 = \int_{y_1=a}^b r(d-c) dy_1 \\
&= [r(d-c)y_1]_a^b = r(d-c)(b-a) \\
r &= \frac{1}{(d-c)(b-a)}
\end{aligned}$$

而且，这也是直观的， r 是全概率1除以区间的面积，随机变量必须落入面积 $(d-c)(b-a)$ 内。

顺便说，式(2-23)说明了如何通过对适当的分布进行采样（随机变量的实现）来估计期望。我们将通过采样和解析计算 y^2 的期望值来说明这个方法。解析解由下式给出：

$$\begin{aligned}
E_{p(y)}\{y^2\} &= \int_{y=a}^b y^2 p(y) dy = \int_{y=a}^b \frac{y^2}{b-a} dy \\
&= \left[\frac{y^3}{3(b-a)} \right]_a^b = \frac{b^3 - a^3}{3(b-a)}
\end{aligned}$$

代入 $a=0$ 、 $b=1$ ，得：

$$E_{p(y)}\{y^2\} = \frac{1}{3}$$

为了计算基于采样的近似值，需要从 $\mathcal{U}(0, 1)$ 得到样本。在 Matlab 中，命令 `rand` 能够从这个分布中生成样本。如果生成 S 个样本 y_s ，我们能够按下式估计期望值：

$$E_{p(y)}\{y^2\} = \frac{1}{S} \sum_{s=1}^S y_s^2 \quad (2-25)$$

图 2-5 表明，当采样数量从 1 增加 10^4 时，近似值改善的程度。真实值 $1/3$ 用虚线表示 (MATLAB 脚本: `approx_expected_value.m`)。只有在 100 个样本后，近似值是相当不错的。使用采样得到近似期望值将会在今后的章节广泛使用 (见练习 EX 2.4)。

2.5.2 β 密度函数

β 密度函数可以用于 $0 \sim 1$ 之间的连续随机变量。 β 密度函数定义如下：

$$p(r) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1} (1-r)^{\beta-1} \quad (2-26)$$

其中 α 、 β 是控制概率密度函数形状的参数，两者都必须为正值。 $\Gamma(z)$ 是伽马函数，此处我们不做讨论，因为在 MATLAB 中我们可以使用内置函数 `gamma` 得到。图 2-6 展示了不同参数下的 β 概率密度函数。我们在第 3 章中大量使用 β 概率密度函数，那时再进行更多的讨论。

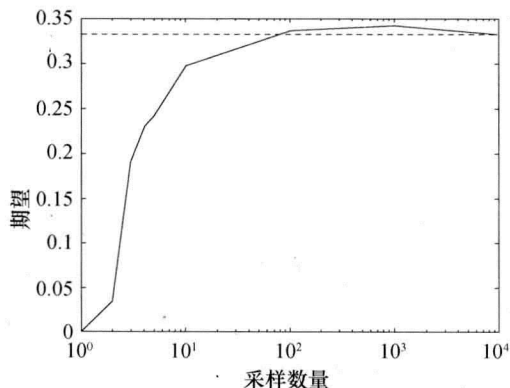


图 2-5 已知式 (2-25)，其中 $p(y) = \mathcal{U}(0,1)$ ，增加采样数量对估计期望的影响。虚线是真实值 ($1/3$)。注意 x 轴按对数缩小。

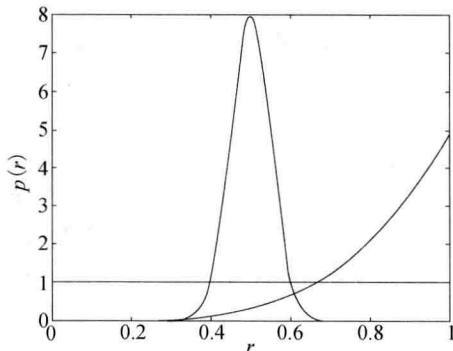


图 2-6 具有 3 对不同参数的 β 概率密度函数的示例

2.5.3 高斯密度函数

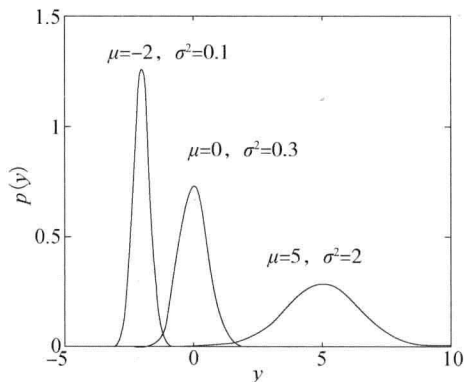
高斯随机变量在许多连续应用中使用。在某些有用的情况下，高斯概率密度函数非常容易操控。高斯分布定义在实数域上（即 $-\infty \sim +\infty$ ）。随机变量 Y 的高斯概率密度函数定义为：

$$p(y|\mu, \sigma^2) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(y-\mu)^2\right\} \quad (2-27)$$

它由两个参数确定——均值（ μ ）和方差（ σ^2 ）。图 2-7 展示了不同参数 μ 、 σ^2 的高斯密度函数。当 $y=\mu$ 时，概率密度函数取最大值，并且概率密度函数关于此点对称。概率密度函数的宽度由参数 σ^2 控制，值越大，密度越宽。如果我们使用图 2-7 最左边的高斯概率密度函数来生成随机变量的实例，那么我们只能期望得到 -2 附近小范围的值。对于最右侧的高斯概率密度函数，我们将获得 5 周围较大范围的值。高斯概率密度函数常用的速记法 $\mathcal{N}(\mu, \sigma^2)$ 表示。因此，如果 Y 服从高斯概率密度函数，那么我们可以记作：

$$p(y|\mu, \sigma^2) = \mathcal{N}(\mu, \sigma^2)$$

读作随机变量 Y 的概率密度函数是正态均值 μ 和方差 σ^2 （高斯和正态通常是可交换的）。图 2-7 不同均值和方差的高斯概率密度函数



2.5.4 多元高斯

高斯分布也可以用于定义连续向量的概率密度函数。向量 $\mathbf{x} = [x_1, \dots, x_D]^T$ 的多元高斯概率密度函数在后续章节中有大量的应用。其概率密度函数定义为：

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \quad (2-28)$$

其中 $\boldsymbol{\mu}$ 是向量（大小与 \mathbf{x} 相同），第 d 个元素表示向量第 d 个元素对应的均值。方差变为一个 $D \times D$ 的协方差矩阵。一个图形化的例子或许是最好的方式来感觉参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$ 对概率密度函数的影响。第一个例子在图 2-8 的第一行。这个例子中，两个参数分别是：

$$\boldsymbol{\mu} = [2, 1]^T, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

这是多元高斯只有两个变量（ x_1 和 x_2 ）且 x_1 、 x_2 相互独立的特例。我们注意到协方差矩阵 $\boldsymbol{\Sigma}$ 是单位矩阵 $\boldsymbol{\Sigma} = \mathbf{I}$ 。所以，

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} |\mathbf{I}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{I}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

由于， $\mathbf{I}^{-1} = \mathbf{I}$ （见注解 1.10）允许我们通过这个表达式获得单元高斯密度函数的乘积。从上述表达式开始（已经将 \mathbf{I}^{-1} 替换为 \mathbf{I} ），我们可以将指数内的矩阵积变成 D 个不同元素的和（见练习 EX 2.5）：

$$\begin{aligned} p(\mathbf{x}) &= \frac{1}{(2\pi)^{D/2} |\mathbf{I}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{I}(\mathbf{x}-\boldsymbol{\mu})\right\} \\ &= \frac{1}{(2\pi)^{D/2} |\mathbf{I}|^{1/2}} \exp\left\{-\frac{1}{2} \sum_{d=1}^D (x_d - \mu_d)^2\right\} \end{aligned}$$

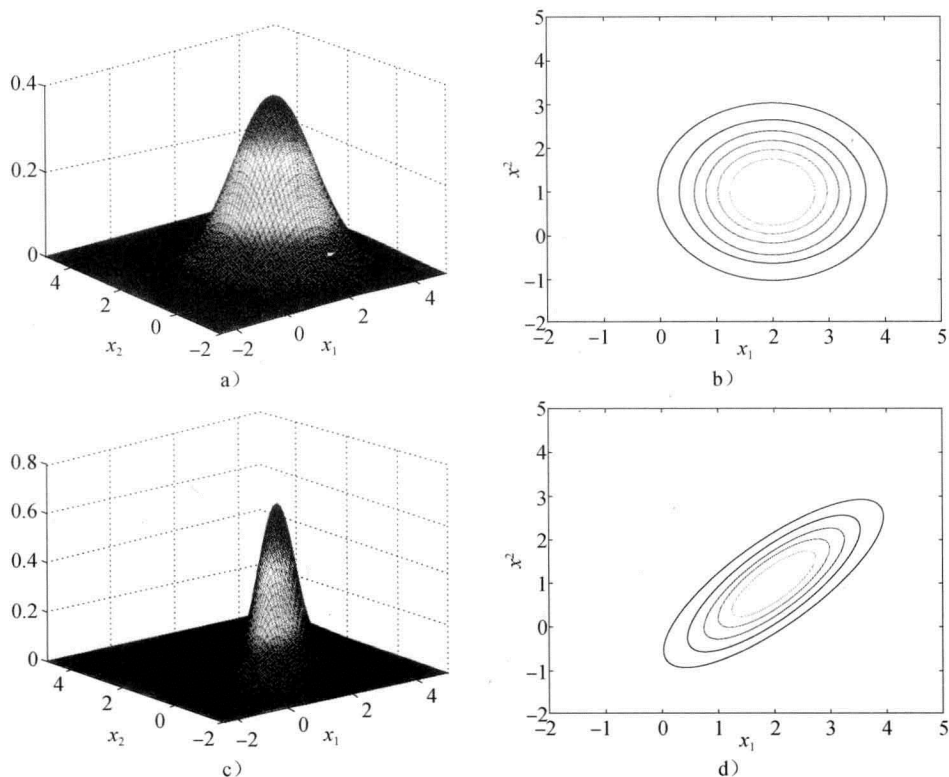


图 2-8 两种不同二元高斯概率密度函数的表面图（左）和等概率率线图（右）

63

注解 2.5 (矩阵的行列式): 方阵的行列式记为 $|\mathbf{A}|$, 对于矩阵 \mathbf{A} , 它是一个很有用的量, 特别是在处理多元高斯概率密度函数时。对于大矩阵, 手动处理行列式太麻烦了, 但是对于小矩阵可以这么做。例如, 对于一个 2×2 的矩阵

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, |\mathbf{A}| = ad - bc$$

但是对于任何比这个矩阵大的矩阵, 最安全的是使用计算机来处理, 除非这是一个特殊结构的矩阵。我们经常看到的一个特殊的矩阵是只有对角线元素的方阵 (对角线以外的所有元素都为 0)。这时, 这个矩阵的行列式仅仅是这些元素的乘积。

例如,

$$\mathbf{A} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{DD} \end{bmatrix}, |\mathbf{A}| = \prod_{d=1}^D a_{dd}$$

要获得行列式表示什么的直观感受并不容易。它在多元高斯归一化常数中的作用与高斯非归一化的体积相关 (归一化体积必须等于 1), 这是非常有用的。

由于和的指数等于指数的乘积, 所以可以将上述表达式改写为:

$$= \frac{1}{(2\pi)^{D/2} |\mathbf{I}|^{1/2}} \prod_{d=1}^D \exp\left\{-\frac{1}{2}(x_d - \mu_d)^2\right\}$$

其中, $|\mathbf{I}|$ 是矩阵 \mathbf{I} 的行列式。从注解 2.5 关于对角矩阵的讨论中可以知道 $|\mathbf{I}| = 1$ 。另一个

常数项 $(2\pi)^{D/2}$ ，可以记为 $\prod_{d=1}^D (2\pi)^{1/2}$ ，所以表达式可以改写为：

$$p(\mathbf{x}) = \prod_{d=1}^D \frac{1}{(2\pi)^{1/2}} \exp\left\{-\frac{1}{2}(x_d - \mu_d)^2\right\}$$

乘积中的每一项是一个单元高斯分布（均值是 μ_d ，方差是 1），因此依据独立性定义，向量 \mathbf{x} 的每个元素相互独立。这个结论不仅仅适用于 $\Sigma = \mathbf{I}$ ，它还适用于协方差矩阵任意的对角线元素不为零的时候。这些对角线元素是每个单元高斯概率密度函数的方差（见练习 EX 2.5、EX 2.6 作为这种类型的高斯概率密度函数处理的进一步练习）。

图 2-8 的第二行是第二个例子，其参数是：

64

$$\boldsymbol{\mu} = [2, 1]^T, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

这个例子中，我们不能把概率密度函数写成单元高斯概率密度函数的乘积，这意味着向量 \mathbf{x} 的每个元素不是相互独立的。在等概率线图中也能看出它们之间的相关性（见图 2-8 右下角）。如果 x_1, x_2 是独立的，那么 $p(x_2 | x_1)$ 不会随着 x_1 的不同而变化。假想 $x_1 = 3$ ，从图 2-8 中可以看出，当 $x_1 = 3$ 时， x_2 的值聚集在 2 周围。如果 $x_1 = 1$ ，那么 x_2 的值聚集在 0 周围。明显地，在两种情况下，期望 x_2 的值不同，直观地， x_1, x_2 是相关的。（MATLAB 脚本：gauss_surf.m）。使用协方差矩阵中的值进行实验可以看出，对表面图和等概率线图的影响。

一个多元高斯很好的特征是它的条件概率密度函数 $p(x_2 | x_1)$ 是另一个很容易得到其均值和方差的高斯概率密度函数。虽然此处我们忽略了细节，但是这是我们经常使用的。

2.5.5 小结

至此，我们完成了对随机变量及其概率的简单介绍。虽然我们仅仅浏览了这个巨大主题的表面内容，但是在前面章节介绍的内容足以使我们模型扩展到显式地度量预测和测量之间的差异。在本章的剩余章节中，我们将在模型中引入一个新的随机变量来对线性模型和数据之间的误差建模。假设随机变量服从高斯密度，对 \mathbf{w} （最优参数值）而言，我们将以与第 1 章中完全一样的方程结束。然而，噪声项的引入将允许我们获得参数值和预测的置信程度。

2.6 产生式的考虑（续）

现在，我们有足够关于随机变量的背景知识来控制线性模型的误差（如图 2-1 所示）。在 2.1.1 节，我们开始思考如何生成与我们观察到的数据相类似的数据。尤其是我们考虑通过形如 $\mathbf{w}^T \mathbf{x}_n$ 的公式产生第 n 次获胜的时间，然后加上一个随机变量 $\epsilon_n - a$ 。

现在，我们的模型采用下列形式：

$$t_n = \mathbf{w}^T \mathbf{x}_n + \epsilon_n \quad (2-29)$$

为了完整定义这个模型，我们需要确定 ϵ_n 的分布。首先，应当清楚模型与实际获胜时间之间的差是一个连续变量。因此， ϵ_n 是一个连续随机变量。我们的确不只有一个随机变量，但是对于每一年奥运会比赛的观测却只有一个。这似乎可以合理地假设，这些值是独立的：

65

$$p(\epsilon_1, \dots, \epsilon_N) = \prod_{n=1}^N p(\epsilon_n)$$

最后的假设是 $p(\epsilon_n)$ 的形式。我们假设这是一个高斯（或正态）分布，其均值为 0、方差为 σ^2 。我们不会做出很多努力来证明这一假设，这里只是允许 ϵ_n 的取值正、负均可（使得数据可以分布于直线 $\mathbf{w}^T \mathbf{x}$ 的上下）并且具有与第 1 章用到的平方损失（squared loss）相

关的有趣的模型属性。正如在 1.1.3 节讨论的关于损失函数的选择问题，在真实的模型中这个选择需要仔细地经过适当的验证。

使用均值 (μ) 为 0, $\sigma^2 = 0.05$ (这里先不用担心这个特殊值) 的正态密度的 ϵ , 即 $p(\epsilon) = \mathcal{N}(\mu, \sigma^2)$ (参见 2.5.3 节), 得到一个更加符合实际的数据集, 详见图 2-9 (MATLAB 脚本: genolymp.m)。

我们的模型现在由两部分组成:

1) 决定性的组部分 ($w^T x_n$), 有时候是指一种趋势或倾向。

2) 随机组部分 (ϵ_n), 有时候是指噪声。

我们已经指出, 我们并非限定噪声为高斯分布, 我们同样也不限定为可加性噪声。在某些应用中, 可乘性可能更加恰当 (此时, $t = f(x; w)\epsilon$)。

例如, 图像像素的退化常常建模为一个具有可乘性噪声的模型。然而, 正如我们在 2.7 节将看到的那样, 选择可加性高斯噪声使我们能够获得最优参数 \hat{w} 的精确表达模式。

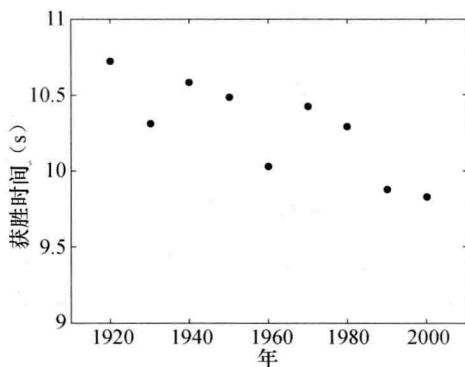


图 2-9 由高斯误差的线性模型生成的数据集

66

2.7 似然估计

我们的模型如下所示:

$$t_n = f(x_n; w) + \epsilon_n \quad \epsilon_n \sim \mathcal{N}(0, \sigma^2)$$

如第 1 章所述, 需要找到 w 的最优值 \hat{w} 。我们还有一个可加性参数 σ^2 需要设置。在第 1 章中, 我们找到了损失最少的 w 值。该损失描述观测值 t 与模型预测值之间的差距。给模型增加一个随机变量的作用是使模型的输出 t 自身也是一个随机变量。换句话说, 对于一个特殊的 x_n , t_n 的值不止一个。因此, 我们不能用损失作为优化 w 和 σ^2 的均值。

给高斯随机变量添加一个常量 ($w^T x_n$) 等同于具有相同常量转换来的均值的另一个高斯随机变量:

$$\begin{aligned} y &= a + z \\ p(z) &= \mathcal{N}(m, s) \\ p(y) &= \mathcal{N}(m + a, s) \end{aligned}$$

因此, 随机变量 t_n 具有如下的密度函数:

$$p(t_n | x_n, w, \sigma^2) = \mathcal{N}(w^T x_n, \sigma^2)$$

注意左边的条件—— t_n 的密度依赖于特定的 x_n 和 w (它们决定均值) 以及 σ^2 (方差) 的值。

为了说明我们如何找到 w 和 σ^2 的最优值, 考虑数据集中一个年份——1980。基于第 1 章的模型 (w_0, w_1), 且再次假设 $\sigma^2 = 0.05$, 我们就可以画出 $p(t_n | x_n = 1980, w, \sigma^2)$ 关于 t_n 的图像, 如图 2-10 所示。其中实线表明:

$$p(t_n | x_n = [1, 1980]^T,$$

$$w = [36.416, -0.0133]^T, \sigma^2 = 0.05)$$

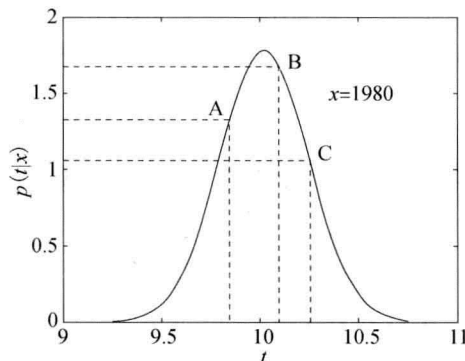


图 2-10 1980 年的似然函数

它是均值 $\mu = 36.416 - 0.0133 \times 1980 = 10.02$ 和方差 $\sigma^2 = 0.05$ 的高斯密度。需要记住的是, 连续随机变量 t 、 $p(t)$ 不能解释为概率。曲线在特定 t 值下的高度可以解释为在 $x = 1980$ 时

观察到这个特定 t 的可能性。1980 年最可能的获胜时间是 10.02 秒（对于高斯分布，最可能的（最高）点代表均值）。图中显示了 3 个时间的例子——A、B 和 C。其中，B 是最可能的，C 是最不可能的。

1980 年奥运会的实际获胜时间是 C（10.25 秒）。作为第 n 年数据的似然值，在 $t_n = 10.25$ 时估算的密度 $p(t_n | \mathbf{x}_n, \mathbf{w}, \sigma^2)$ 是一个重要值。我们不能改变 $t_n = 10.25$ （因为这是我们的数据），但是我们可以改变 \mathbf{w} 和 σ^2 来尝试并移动密度，使其在 $t_n = 10.25$ 时尽可能保持高的可能性。这种通过寻找参数以最大化似然值的方式是机器学习中的一个重要观点。

2.7.1 数据集的似然值

一般来说，我们感兴趣的并非是单个数据点的似然值，而是整个数据集所有点的似然值。如果有 N 个数据点，我们感兴趣的就是它们的联合条件密度：

$$p(t_1, \dots, t_N | \mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{w}, \sigma^2)$$

这是数据集中所有点的联合密度（参见 2.2.5 节）。我们将利用第 1 章定义的向量表示法和 \mathbf{X} ，将其改写为紧缩形式为 $p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \sigma^2)$ 。估计观测值的这种密度，产生了所有数据集的单个似然值，它可以通过改变 \mathbf{w} 和 σ^2 来进行优化。

所有数据点的噪声是独立的（ $p(\epsilon_1, \dots, \epsilon_N) = \prod_n p(\epsilon_n)$ ）这一假设使我们能够将这种密度分解为更易操作的对象。特别地，这一联合密度可以分解为 N 个独立的部分，每一部分对应一个数据对象：

$$L = p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^N p(t_n | \mathbf{x}_n, \mathbf{w}, \sigma^2) = \prod_{n=1}^N \mathcal{N}(\mathbf{w}^T \mathbf{x}_n, \sigma^2) \quad (2-30)$$

注意，我们还没有说 t_n 自身也是完全独立的。它不是这种情况——平均看， t_n 随时间增加，表明了它们之间清晰的统计依赖性。如果它们完全独立，那么就根本不值得对数据进行建模。事实上，它们是条件独立的——对于给定的 \mathbf{w} 值（模型的决定性部分）， t_n 是独立的，没有此条件则不独立。如果这听起来有点奇怪，还可以这样认为：假设我们搜集所有奥运会的年代及获胜的时间，除了中间的某一年——如 1960 年。为了简便起见，我们使用 \mathbf{X} 、 \mathbf{t} 代表除了 1960 年外所有奥运会的年代和获胜时间。如果我们试图使用 \mathbf{X} 和 \mathbf{t} 来学习 t_{1960} ，那么我们感兴趣的是如下所示的条件分布：

$$p(t_{1960} | \mathbf{x}_{1960}, \mathbf{X}, \mathbf{t})$$

根据条件分布的定义，可以给出如下等式：

$$p(t_{1960} | \mathbf{x}_{1960}, \mathbf{X}, \mathbf{t}) = \frac{p(t_{1960}, \mathbf{t} | \mathbf{x}_{1960}, \mathbf{X})}{p(\mathbf{t} | \mathbf{X})}$$

假设参数 \mathbf{t} 的元素是独立的，可得出 t_{1960} 仅依赖于 \mathbf{x}_{1960} ：

$$p(t_{1960} | \mathbf{x}_{1960}, \mathbf{X}, \mathbf{t}) = \frac{p(t_{1960} | \mathbf{x}_{1960}) \prod_n p(t_n | \mathbf{x}_n)}{\prod_n p(t_n | \mathbf{x}_n)} = p(t_{1960} | \mathbf{x}_{1960})$$

然而，为了使模型适用于任何应用， t_{1960} 在某种程度上必须依赖于其他数据。这种依赖性包括到参数 \mathbf{w} 中。模型的决定性部分捕获了这种依赖性。如果已知 \mathbf{w} ，那么剩下的就是观测数据与 $\mathbf{w}^T \mathbf{x}_n$ 之间的差值。假设误差是独立的，因此以 \mathbf{w} 为条件，观测值也是独立的。没有一个模型，其观测值不是独立的。

接下来，我们将说明如何找到 \mathbf{w} 和 σ^2 的值来最大化似然值。

2.7.2 最大似然

在已知当前模型(即选定的 \mathbf{w} 和 σ^2)情况下,式(2-30)给出了一个数值来表示数据集的相似性。由于数据集是固定的,所以不断变化的模型将产生不同的似然值。模型的一个明智选择旨在最大化似然值。换句话说,我们将选择那些能够使观测值最相似的模型参数值。

出于统计的原因,我们将最大化似然值的**自然对数值**(我们将按照机器学习的惯例,使用 $\log(y)$ 来表示 y 的自然对数,而其他地方常常用 $\ln(y)$ 表示)。我这样做,因为估计的参数 $\hat{\mathbf{w}}$ 和 $\hat{\sigma}^2$ 在最大化似然对数时,同样最大化了似然值。

取代高斯密度函数的表示形式(式(2-27))并分离各个变量,就得到了更加详细的表示形式:

$$\begin{aligned}\log L &= \sum_{n=1}^N \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} (t_n - f(\mathbf{x}_n; \mathbf{w}))^2 \right\} \right) \\ &= \sum_{n=1}^N \left(-\frac{1}{2} \log(2\pi) - \log \sigma - \frac{1}{2\sigma^2} (t_n - f(\mathbf{x}_n; \mathbf{w}))^2 \right) \\ &= -\frac{N}{2} \log 2\pi - N \log \sigma - \frac{1}{2\sigma^2} \sum_{n=1}^N (t_n - f(\mathbf{x}_n; \mathbf{w}))^2\end{aligned}$$

用 $f(\mathbf{x}_n; \mathbf{w}) = \mathbf{w}^T \mathbf{x}_n$ 替换模型中的决定性部分,对数似然表达式就呈现如下的形式:

$$\log L = -\frac{N}{2} \log 2\pi - N \log \sigma - \frac{1}{2\sigma^2} \sum_{n=1}^N (t_n - \mathbf{w}^T \mathbf{x}_n)^2 \quad (2-31)$$

至于第1章得到的最小二乘解,通过求导数、使其等于零以及求解拐点的方法,能够找到最优参数,这类似于1.1.4节所述的方式。对于 \mathbf{w} (注意, $\mathbf{w}^T \mathbf{x}_n = \mathbf{x}_n^T \mathbf{w}$),

$$\begin{aligned}\frac{\partial \log L}{\partial \mathbf{w}} &= \frac{1}{\sigma^2} \sum_{n=1}^N \mathbf{x}_n (t_n - \mathbf{x}_n^T \mathbf{w}) \\ &= \frac{1}{\sigma^2} \sum_{n=1}^N \mathbf{x}_n t_n - \mathbf{x}_n \mathbf{x}_n^T \mathbf{w} = \mathbf{0}\end{aligned}$$

注意, $\frac{\partial \log L}{\partial \mathbf{w}}$ 是向量,因此我们令其为 $\mathbf{0}$,即一个相同大小的为零的向量。记得第1章采用的简写矩阵/向量:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}, \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}$$

在该表达式中, $\sum_{n=1}^N \mathbf{x}_n t_n$ 可以写成 $\mathbf{X}^T \mathbf{t}$,同理, $\sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^T \mathbf{w}$ 可以写成 $\mathbf{X}^T \mathbf{X} \mathbf{w}$ (参见练习EX 1.5)。这就允许我们用更方便的向量/矩阵形式写出其导数:

$$\frac{\partial \log L}{\partial \mathbf{w}} = \frac{1}{\sigma^2} (\mathbf{X}^T \mathbf{t} - \mathbf{X}^T \mathbf{X} \mathbf{w}) = \mathbf{0} \quad (2-32)$$

解该表达式中的 \mathbf{w} ,可以得到最优值的表达式:

$$\begin{aligned}\frac{1}{\sigma^2} (\mathbf{X}^T \mathbf{t} - \mathbf{X}^T \mathbf{X} \mathbf{w}) &= \mathbf{0} \\ \mathbf{X}^T \mathbf{t} - \mathbf{X}^T \mathbf{X} \mathbf{w} &= \mathbf{0} \\ \mathbf{X}^T \mathbf{X} \mathbf{w} &= \mathbf{X}^T \mathbf{t} \\ \mathbf{w} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}\end{aligned}$$

这就是 \mathbf{w} 的最大似然解:

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t} \quad (2-33)$$

值得注意的是，该解是正确的，它已经在第1章中（1.16节）通过最小二乘法得到。如果将噪声假设为高斯分布，则最小化平方损失等同于最大似然解。同样，噪声变量 σ^2 ，在该表达式中并未出现——它衡量似然值，但它不会影响 $\hat{\mathbf{w}}$ 对应的最大值。

为了获得 σ^2 的表达式（假设 $\mathbf{w} = \hat{\mathbf{w}}$ ），我们可以采用相同的步骤。采用偏导数和令其等于0，可以得到：

$$\frac{\partial L}{\partial \sigma} = -\frac{N}{\sigma} + \frac{1}{\sigma^3} \sum_{n=1}^N (t_n - \mathbf{x}^T \hat{\mathbf{w}}) = 0 \quad (2-34)$$

70

重排给定的 $\hat{\sigma}^2$ ， σ^2 的最大似然估计为：

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N (t_n - \mathbf{x}^T \hat{\mathbf{w}})^2 \quad (2-35)$$

这个表达式非常有意义——变量就是简单的均方误差。我们更喜欢用矩阵表示，因此鉴于事实 $\sum_{n=1}^N (t_n - \mathbf{x}^T \hat{\mathbf{w}})^2$ 等于 $(\mathbf{t} - \mathbf{X} \hat{\mathbf{w}})^T (\mathbf{t} - \mathbf{X} \hat{\mathbf{w}})$ ，

$$\begin{aligned} \sigma^2 &= \frac{1}{N} (\mathbf{t} - \mathbf{X} \hat{\mathbf{w}})^T (\mathbf{t} - \mathbf{X} \hat{\mathbf{w}}) \\ &= \frac{1}{N} (\mathbf{t}^T \mathbf{t} - 2 \mathbf{t}^T \mathbf{X} \hat{\mathbf{w}} + \hat{\mathbf{w}}^T \mathbf{X}^T \mathbf{X} \hat{\mathbf{w}}) \end{aligned} \quad (2-36)$$

这还可以借助 $\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}$ 进行进一步简化（注意，因为 $(\mathbf{X}^T \mathbf{X})^{-1}$ 是对称的，所以 $\hat{\mathbf{w}}^T = \mathbf{t}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1}$ ，因此它等于其自身的转置）：

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{N} (\mathbf{t}^T \mathbf{t} - 2 \mathbf{t}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t} + \mathbf{t}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}) \\ &= \frac{1}{N} (\mathbf{t}^T \mathbf{t} - 2 \mathbf{t}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t} + \mathbf{t}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}) \\ &= \frac{1}{N} (\mathbf{t}^T \mathbf{t} - \mathbf{t}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}) \end{aligned}$$

$$\hat{\sigma}^2 = \frac{1}{N} (\mathbf{t}^T \mathbf{t} - \mathbf{t}^T \mathbf{X} \hat{\mathbf{w}}) \quad (2-37)$$

利用奥运会100米数据，最优的参数值（等于1阶（线性）多项式）为：

$$\hat{\mathbf{w}} = [36.4165, -0.0133]^T, \hat{\sigma}^2 = 0.0503$$

$\hat{\mathbf{w}}$ 等同于第1章得到的最小二乘解（它们使用相同的表达式求得）。 $\hat{\sigma}^2$ 说明了高斯噪声的方差，我已经假设其用于破坏我们的数据。本章的后面我们将看到，这样对噪声建模有利于损失的最小化。在此之前，我们先看看解的特点。

2.7.3 最大似然解的特点

在第1章中，我们使用损失函数的2阶导数来确保我们已经找到最小值。现在我们将同样用似然的2阶导数来确保我们已经找到最大的似然值。我们的2阶导数现在是关于向量的并测试2阶导数，我们构建了 **Hessian 矩阵**（参见注解2.6）。该矩阵中的每个元素都是关于 \mathbf{w} 元素对的2阶导数。为了确保我们已经找到了最大似然值，我们必须证实海 Hessian 阵是负定的（参见注解2.7）。

71

注解 2.6 (Hessian 矩阵): Hessian 矩阵是一个包含所有函数的 2 阶偏导数的方阵。例如, 带有参数 $\mathbf{w} = [w_1, \dots, w_K]^T$ 的函数 $f(\mathbf{x}, \mathbf{w})$ 的 Hessian 矩阵为:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial w_1^2} & \frac{\partial^2 f}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 f}{\partial w_1 \partial w_K} \\ \frac{\partial^2 f}{\partial w_2 \partial w_1} & \frac{\partial^2 f}{\partial w_2^2} & \dots & \frac{\partial^2 f}{\partial w_2 \partial w_K} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial w_K \partial w_1} & \frac{\partial^2 f}{\partial w_K \partial w_2} & \dots & \frac{\partial^2 f}{\partial w_K^2} \end{bmatrix}$$

我们可以从 Hessian 矩阵中了解方程 $f(\mathbf{x}; \mathbf{w})$ 中的拐点信息。例如, 如果 Hessian 矩阵在某一拐点 $\hat{\mathbf{w}}$ 是负定的, 那么我们知道这一拐点就是极大值。

注解 2.7 (负定矩阵): 如果一个实矩阵 \mathbf{H} 对于所有的实值向量 \mathbf{x} 满足

$$\mathbf{x}^T \mathbf{H} \mathbf{x} < 0$$

则称该矩阵是负定的。

2 阶偏导数的 Hessian 矩阵可以通过对式 (2-32) 关于 \mathbf{w}^T 求导数而求解:

$$\frac{\partial^2 \log L}{\partial \mathbf{w} \partial \mathbf{w}^T} = -\frac{1}{\sigma^2} \mathbf{X}^T \mathbf{X} \quad (2-38)$$

如果用 $\mathbf{x}_n = [1, x_n]^T$ 进行替换, 那么该矩阵的对角线元素就等于 (它们的不同之处在于乘以一个常数) 式 (1-9) 的 2 阶偏导数 (参见练习 EX 2.7)。

这肯定是一个最大值, 我们需要确定这个矩阵是否是负定的。我们可以通过如下来实现:

$$-\frac{1}{\sigma^2} \mathbf{z}^T \mathbf{X}^T \mathbf{X} \mathbf{z} < 0$$

对于任意向量 \mathbf{z} 或者相当于 (因为 σ^2 必须为正的):

$$\mathbf{z}^T \mathbf{X}^T \mathbf{X} \mathbf{z} > 0$$

到此, 证实如何做到这一点是有价值的。我们假设每个 x_n 是二维的, 这样即可展开得到各项。更一般地, 我们将 \mathbf{X} 定义为与以前稍微不同:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{N1} & x_{N2} \end{bmatrix}$$

因此, $\mathbf{X}^T \mathbf{X}$ 为

$$\mathbf{X}^T \mathbf{X} = \begin{bmatrix} \sum_{i=1}^N x_{i1}^2 & \sum_{i=1}^N x_{i1} x_{i2} \\ \sum_{i=1}^N x_{i2} x_{i1} & \sum_{i=1}^N x_{i2}^2 \end{bmatrix}$$

它与一个随机实向量 $\mathbf{z} = [z_1, z_2]^T$ 进行前乘或后乘, 结果为:

$$\begin{aligned} \mathbf{z}^T \mathbf{X}^T \mathbf{X} \mathbf{z} &= \left[z_1 \sum_{i=1}^N x_{i1}^2 + z_2 \sum_{i=1}^N x_{i2} x_{i1}, z_1 \sum_{i=1}^N x_{i1} x_{i2} + z_2 \sum_{i=1}^N x_{i2}^2 \right] \mathbf{z} \\ &= z_1^2 \sum_{i=1}^N x_{i1}^2 + 2z_1 z_2 \sum_{i=1}^N x_{i1} x_{i2} + z_2^2 \sum_{i=1}^N x_{i2}^2 \end{aligned}$$

由于第一项和最后一项必须是正的，所以证明该表达式大于0就等同于证明它们的和大于中间项：

$$z_1^2 \sum_{i=1}^N x_{i1}^2 + z_2^2 \sum_{i=1}^N x_{i2}^2 > 2z_1 z_2 \sum_{i=1}^N x_{i1} x_{i2}$$

定义 $y_{i1} = z_1 x_{i1}$ ， $y_{i2} = z_2 x_{i2}$ ，并将其代入上表达式，得：

$$\sum_{i=1}^N (y_{i1}^2 + y_{i2}^2) > 2 \sum_{i=1}^N y_{i1} y_{i2}$$

现在，对于任意的 i ，

$$\begin{aligned} y_{i1}^2 + y_{i2}^2 &> 2y_{i1} y_{i2} \\ y_{i1}^2 - 2y_{i1} y_{i2} + y_{i2}^2 &> 0 \\ (y_{i1} - y_{i2})^2 &> 0 \end{aligned}$$

只有当 $y_{i1} = y_{i2}$ ，故 $x_{i1} = x_{i2}$ 时该式才不成立——该情况在实际中不可能发生。因此，如果对于任意 i ， $y_{i1}^2 + y_{i2}^2 > 2y_{i1} y_{i2}$ 成立，则任意数目的该项的和也一定满足该不等式。因此 $\mathbf{z}^T \mathbf{X}^T \mathbf{X} \mathbf{z}$ 恒大于零，Hessian 矩阵是负定的，且解就是最大似然值。

为了确保 $\hat{\sigma}^2$ 就是最大似然值，我们对式 (2-24) 关于 σ 求导数：

$$\frac{\partial^2 \log L}{\partial \sigma^2} = \frac{N}{\sigma^2} - \frac{3}{\sigma^4} (\mathbf{t} - \mathbf{X} \hat{\mathbf{w}})^T (\mathbf{t} - \mathbf{X} \hat{\mathbf{w}})$$

我们可以替换式 (2-36) 所给的 $\hat{\sigma}^2$ 值来简化该表达式，结果为：

$$\begin{aligned} \frac{\partial^2 \log L}{\partial \sigma^2} &= \frac{N}{\hat{\sigma}^2} - \frac{3}{(\hat{\sigma}^2)^2} N \hat{\sigma}^2 \\ &= -\frac{2N}{\hat{\sigma}^2} \end{aligned}$$

73

上式恒小于零，因此 $\hat{\sigma}^2$ 为最大似然值。

2.7.4 最大似然法适用于复杂模型

将 $\hat{\sigma}^2$ 的表达式 (式 (2-35)) 代入对数似然表达式 (式 (2-31))，得到了最大值的对数似然值：

$$\begin{aligned} \log L &= -\frac{N}{2} \log 2\pi - \frac{N}{2} \log \hat{\sigma}^2 - \frac{1}{2\hat{\sigma}^2} N \hat{\sigma}^2 \\ &= -\frac{N}{2} (1 + \log 2\pi) - \frac{N}{2} \log \hat{\sigma}^2 \end{aligned}$$

该式告诉我们， L 的最大值随着 $\hat{\sigma}^2$ 的减小而增大。 σ^2 是噪声的方差，而噪声是模型的组成部分，其目的在于捕获模型（即 $f(\mathbf{x}; \mathbf{w})$ ）中决定性部分不能捕获的影响。减小 σ^2 的一种方法是调整 $f(\mathbf{x}; \mathbf{w})$ 使其难以捕获数据中更多的变异性——使其更加灵活。例如，对于奥运会 100 米数据，可以通过拟合阶数不断增大的多项式来增加似然值，从而研究模型的灵活性（或复杂性）。图 2-11a 表明 $\log L$ 随着多项式阶数的增大而增大符合奥运会 100 米数据 (MATLAB 脚本: `olymplike.m`)。如果我们打算用 $\log L$ 帮助我们选择使用哪个特定的模型，那么它总是倾向于选择那些复杂度不断增大的模型。这看起来似乎是一个明智的决策——因为随着 $\hat{\sigma}^2$ 的减小，模型能够捕获数据中更多的变异性。然而，考虑如果我们的任务是预测我们还没观察到的某一年的获胜时间（比如 2016 年）。图 2-11b 显示了 1 阶（虚线）和 8 阶（实线）多项式均适用于预测 2016 年的数据（用大的黑点表示）。更复杂的模型

预测获胜的时间接近 11 秒（这可能是至今最慢的速度），而简单模型的预测更真实。对人来说，似乎是简单模型捕获了数据中重要的关系（不断下降的趋势）而更复杂的模型却没有。这就是 1.5 节所看见的变化和过度拟合之间取舍的一个漂亮例子。简单模型比复杂模型具有更好的概括性。复杂模型过度拟合——我们给了其更大的空间以至于它试图将本质上是噪声的数据变得有意义。在 1.6 节，我们已经见证了如何将正则化应用于惩罚过度复杂的参数值。该方法同样可以通过参数值的先验分布用于概率模型，这将在第 3 章介绍。

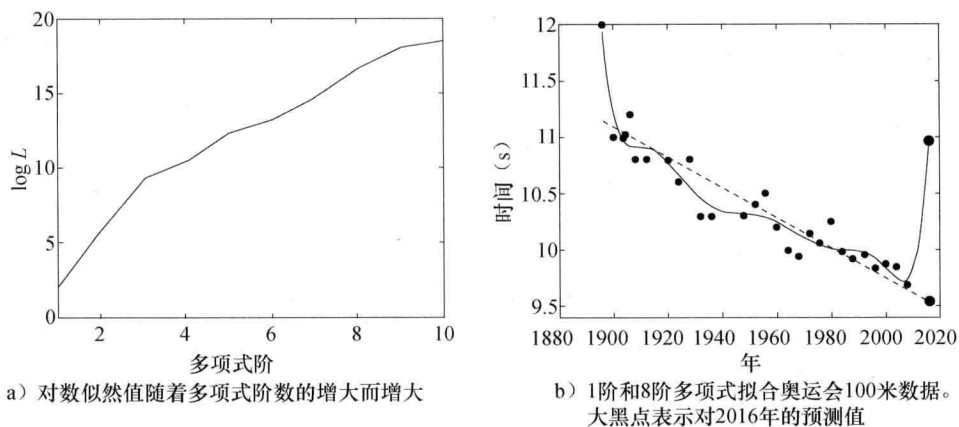


图 2-11 关于奥运会男子 100 米数据模型复杂度的实例

2.8 偏差-方差平衡问题

1.5 节已经讨论过的泛化与过度拟合的折中问题有时描述为偏差-方差的折中问题。试想我们已经获得了我们取样数据的分布 $p(\mathbf{x}, t)$ 。理论上，我们可以使用这个分布来计算估计参数值和真实值之间期望的均方误。我们希望这个值 $\overline{\mathcal{M}}$ ，尽可能地小。它是由偏差 \mathcal{B} 与方差 \mathcal{V} 两部分组成：

$$\overline{\mathcal{M}} = \mathcal{B}^2 + \mathcal{V}$$

偏差-方差平衡描述的是模型和生成数据过程的系统误差。模型越简单，其偏离程度越高（低拟合）。因此我们可以通过减小偏差和对 $\overline{\mathcal{M}}$ 的贡献来使模型更复杂。遗憾的是，越复杂的模型具有越大的方差，因此增大了 $\overline{\mathcal{M}}$ 中的 \mathcal{V} 值。寻找泛化和低/过度拟合间的平衡问题也就是寻找偏差-方差平衡问题。

我们略掉了进一步的详细介绍，但是更细节的内容可以在本章结尾推荐的读物中找到。

2.8.1 小结

在 2.7 节，我们已经介绍了很多新概念。首先我们介绍了一个显式地对数据噪声（或误差）建模的例子。通过假设这些误差可以用高斯随机变量进行建模，我们证实我们可以计算描述数据如何相似的称为似然值的量。假设参数服从高斯分布，当已知最优化参数的相同的表达式选择参数和最大化似然函数最小化平方误差时，这是一个合理的值。在本章的剩余部分中，我们将分析对噪声建模的两个重要好处：能够确定不确定参数的个数和能够表达预测的不确定性。

2.9 噪声对参数估计的影响

在本节中，将会推导出参数估计有多大置信度的表达式——我们如何改变直线但仍然有一个好的模型。如果噪声很大（ σ^2 很高），可能我们就能够忍受 $\hat{\mathbf{w}}$ 大的改变。如果噪声很小，那么拟合的质量就将会快速恶化。在我们得到表达式前，有必要探究在生成综合数据中

\hat{w} 的变化程度。特别地，我们将生成大量具有相同真实 w 和 σ^2 的数据，来看看最大似然值如何估计 \hat{w} 的变化。参考如下模型：

$$t_n = w_0 + w_1 x_n + \varepsilon_n \quad \varepsilon_n \sim \mathcal{N}(0, \sigma^2) \quad (2-39)$$

假设真实参数值为 $w_0 = -2, w_1 = 3$ ，且噪声方差 $\sigma^2 = 0.5^2$ ，我们就能够针对特定的一组特征值 $(x_1, x_2, x_3, \dots, x_N)$ 生成我们想要的任意多组数据 $(t_1, t_2, t_3, \dots, t_N)$ ，并且能够计算每一组数据的 \hat{w} 。图 2-12 显示了一个这种数据集的例子和真实方程，其中特征值包含了 20 个 $(0, 1)$ 均匀分布的值，即 $p(x) = \mathcal{U}(0, 1)$ 。图 2-13 显示了生成的 10 000 个数据集以及每一种情况下的拟合值 \hat{w} 。图 2-13a 中的柱形图每个条形的高度表示生成某种特定范围内参数值的数据集的个数；图 2-13b 表示其相同含义的等概率线图。我们可以发现在真实值周围， w_0 和 w_1 均在较大范围内变化。很难从这些值得到该模型所呈现出来的变化程度。图 2-12 式 (2-39) 所示模型生成的数据和真实函数该例中 10 个数据集的 \hat{w} 和真实函数如图 2-14 所示。

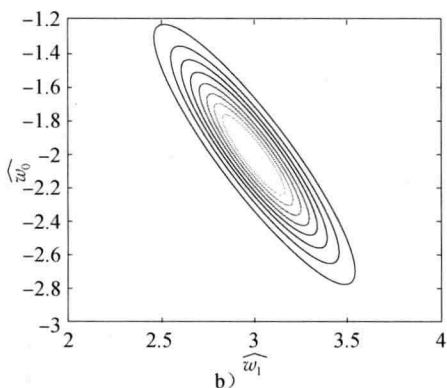
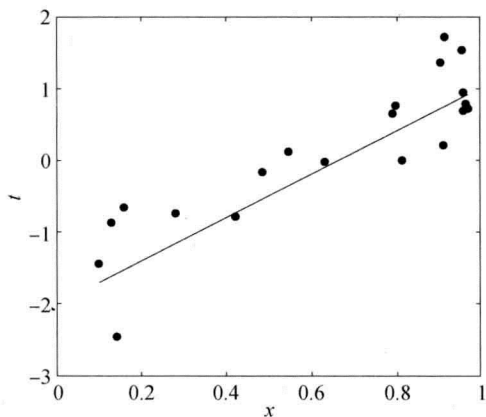


图 2-13 式 (3-39) 所示模型生成的 10 000 个数据集的 \hat{w} 的变化

如果假设真实数据集也是通过同样的过程生成的，那么它对于定量估计结果的变异性很有帮助。遗憾的是，我们没有很多能进行比较 \hat{w} 的数据集。2.9.1 节将介绍如何利用可用数据来确定这种不确定性。

2.9.1 参数估计的不确定性

我们在最后一节说明我们所获得的 \hat{w} 值受到数据中特定噪声的严重影响。鉴于这种情况，它将有损于认识 \hat{w} 的不确定性程度。换句话说， \hat{w} 能够很好地诠释数据的唯一值还是说有很多这样的值能做到同样的效果？

为了更进一步，我们必须弄清楚 w 和 \hat{w} 的含义。我们已经提出了一个与数据有关的模型，这个模型就是：

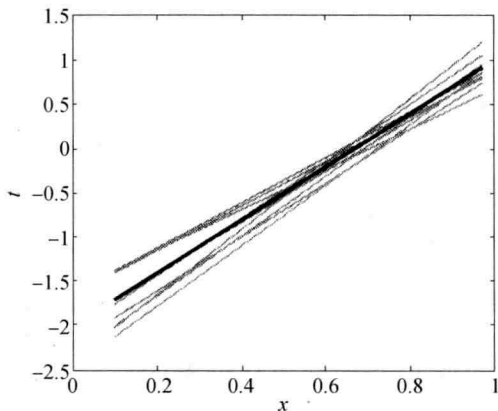


图 2-14 式 (2-39) 所示模型生成的 10 个数据集所推导的函数，即真实函数（较粗较黑的直线）

$$t_n = \mathbf{w}^T \mathbf{x}_n + \varepsilon_n$$

其中 \mathbf{w} 表示参数的真实值, ε_n 是一个已经定义为正态分布的随机变量。这一假设意味着生成分布 (或似然值) $p(t|\mathbf{X}, \mathbf{w}, \sigma^2)$ 是许多正态密度的乘积:

$$p(t|\mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^N p(t_n | \mathbf{x}_n \mathbf{w}, \sigma^2) = \prod_{n=1}^N \mathcal{N}(\mathbf{w}^T \mathbf{x}_n, \sigma^2)$$

在 2.5.4 节中, 我们已经介绍了如何将单元高斯密度的乘积写成一个具有对角协方差的多元高斯密度。用单个多元高斯密度比用多个单元高斯密度的积简洁。这样, 多元高斯分布就是:

$$p(t|\mathbf{X}, \mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2 \mathbf{I})$$

令自己满意的是均值和协方差均是正确的。现在 $\hat{\mathbf{w}}$ 是真实参数值 \mathbf{w} 的估计。通过计算针对生成数据分布的 $\hat{\mathbf{w}}$ 的期望值 (2.2.8 节), 将会告诉我们所期望的 $\hat{\mathbf{w}}$, 其平均值为:

$$E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{\hat{\mathbf{w}}\} = \int \hat{\mathbf{w}} p(t|\mathbf{X}, \mathbf{w}, \sigma^2) dt$$

将 $\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T t$ 代入上式, 我们就能够估计其积分:

$$\begin{aligned} E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{\hat{\mathbf{w}}\} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \int t p(t|\mathbf{X}, \mathbf{w}, \sigma^2) dt \\ E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{\hat{\mathbf{w}}\} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{t\} \\ E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{\hat{\mathbf{w}}\} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X} \mathbf{w} \\ E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{\hat{\mathbf{w}}\} &= \mathbf{w} \end{aligned} \quad (2-40)$$

其中, 用到这样一个事实——正态分布随机变量的期望值等于其均值 ($E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{t\} = \mathbf{X}\mathbf{w}$, 因为 $p(t|\mathbf{X}, \mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2 \mathbf{I})$)。

该结果告诉我们, 逼近 $\hat{\mathbf{w}}$ 的期望值是参数的真实值。本章后面将更加详细地探讨它, 但是它意味着我们的估计值是**无偏的**——这是错误的, 因为这是平均值, 实际可能更大或更小。

$\hat{\mathbf{w}}$ 估计中的变异性包含在它的协方差矩阵中。针对我们的目的, 这个协方差矩阵能够提供两方面有用的信息。对角元素 ($\hat{\mathbf{w}}$ 中单个元素的变异性) 告诉我们单个参数中期望的变异性——即它们被数据定义的好坏程度。在前面的实验中, 参数表现出较大的变异性, 表明它们没有被数据很好地定义。非对角元素告诉我们, 参数如何协同变异——如果值很高且为正, 它就告诉我们增加一个值将导致其他值增大, 从而得到一个非常好的模型。大量负值则告诉我们相反的信息——即增大一个值导致其他值减小。趋近于零的值告诉我们这个参数与其他参数是独立的。例如, 上文提到的例子 (见图 2-13), 似乎是增大 w_1 , 导致 w_0 下降, 因此我们期望协方差矩阵中的非对角元素是负值。

在 2.2.8 节, 我们得到了协方差矩阵的通用表达式 (式 (2-16)), 将 t 和 $p(t|\mathbf{X}, \mathbf{w}, \sigma^2)$ 代入该式并使用之前的结果 $E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{\hat{\mathbf{w}}\} = \mathbf{w}$, 将得到:

$$\begin{aligned} \text{cov}\{\hat{\mathbf{w}}\} &= E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{\hat{\mathbf{w}} \hat{\mathbf{w}}^T\} - E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{\hat{\mathbf{w}}\} E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{\hat{\mathbf{w}}\}^T \\ &= E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{\hat{\mathbf{w}} \hat{\mathbf{w}}^T\} - \mathbf{w} \mathbf{w}^T \end{aligned} \quad (2-41)$$

其中我们使用前面得到的 $\hat{\mathbf{w}}$ 的期望值。为了计算这个值, 将从第一项开始。通过将 $\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T t$ 代入并删掉所有不含 t 的项, 该式可以展开为:

$$\begin{aligned} E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{\hat{\mathbf{w}} \hat{\mathbf{w}}^T\} &= E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T t)((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T t)^T\} \\ &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)} \{t t^T\} \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \end{aligned} \quad (2-42)$$

现在, $p(t|\mathbf{X}, \mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2 \mathbf{I})$ 。因此, 根据定义, t 的协方差是 $\sigma^2 \mathbf{I}$, 均值是 $\mathbf{X}\mathbf{w}$ 。

使用同样的推导方式，将式 (2-41) 变为：

$$\text{cov}\{\mathbf{t}\} = \sigma^2 \mathbf{I} = \mathbf{E}_{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{t}\mathbf{t}^T\} - \mathbf{E}_{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{t}\}\mathbf{E}_{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{t}\}^T \quad (2-43)$$

因此，我们可以重新组织该表达式，得到 $\mathbf{E}_{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{t}\mathbf{t}^T\}$ 的表达式为：

$$\begin{aligned} \mathbf{E}_{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{t}\mathbf{t}^T\} &= \mathbf{E}_{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{t}\}\mathbf{E}_{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{t}\}^T + \sigma^2 \mathbf{I} \\ &= \mathbf{X}\mathbf{w}(\mathbf{X}\mathbf{w})^T + \sigma^2 \mathbf{I} \\ &= \mathbf{X}\mathbf{w}\mathbf{w}^T\mathbf{X}^T + \sigma^2 \mathbf{I} \end{aligned}$$

将该式代入式 (2-42)，将得到：

$$\begin{aligned} \mathbf{E}_{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{w}\mathbf{w}^T\} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\mathbf{w}\mathbf{w}^T\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \\ &\quad + \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \\ &= \mathbf{w}\mathbf{w}^T + \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} \end{aligned} \quad (2-44)$$

最后，将该式代入式 (2-41)，将得到 $\hat{\mathbf{w}}$ 的协方差矩阵的表达式：

$$\begin{aligned} \text{cov}\{\hat{\mathbf{w}}\} &= \mathbf{w}\mathbf{w}^T + \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} - \mathbf{w}\mathbf{w}^T \\ &= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} \end{aligned} \quad (2-45)$$

它是之前得到的 2 阶导数的 Hessian 矩阵式 (2-38) 的负反函数，即

$$\text{cov}\{\hat{\mathbf{w}}\} = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} = -\left(\frac{\partial^2 \log L}{\partial \mathbf{w} \partial \mathbf{w}^T}\right)^{-1} \quad (2-46)$$

该结果告诉我们，参数的确定性/不确定性（用 $\text{cov}(\hat{\mathbf{w}})$ 表示的）直接与对数似然值的 2 阶导数相关。对数似然值的 2 阶导数告诉我们似然函数的弯曲程度。因此，低弯曲对应于参数较高的不确定性，而高弯曲对应于较低的不确定性。换句话说，我们有了一个能够告诉我们数据能够给我们多少关于参数估计的信息表达式。事实上，矩阵 $\sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$ 是一个叫做费舍尔信息 (Fisher Information) 矩阵 (\mathcal{I}) 的负反函数。费舍尔信息矩阵是用对数似然值的 2 阶导数矩阵的期望值来计算的：

$$\mathcal{I} = \mathbf{E}_{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2)}\left\{-\frac{\partial^2 \log p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2)}{\partial \mathbf{w} \partial \mathbf{w}^T}\right\}$$

我们已经知道括号中的项是什么——我们之前计算的 Hessian 矩阵，因此

$$\mathcal{I} = \mathbf{E}_{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \sigma^2)}\left\{\frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X}\right\}$$

由于期望值是一个常数，因此它为

$$\mathcal{I} = \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X} \quad (2-47)$$

\mathcal{I} 告诉我们数据能给我们提供多少关于一个特定参数（对角元素）或参数对（非对角元素）的信息（越是负值，信息所呈现的信息量越大）。直观地讲，如果数据噪声非常大，则信息量就很低。一般来说，如果信息量很大，那么数据能够提供非常准确的参数估计且 $\hat{\mathbf{w}}$ 的协方差将很低 ($\text{cov}(\hat{\mathbf{w}}) = \mathcal{I}^{-1}$)。如果信息量很低，协方差将很高（参见练习 EX 2.13 和 EX 2.14）。

看一个例子，观察图 2-15 中最上面的线。左侧图显示了数据和真实函数 ($t = 3x - 2$)，右侧图显示了两个参数函数的似然值。我们可以发现，由于较高程度的噪声，所以似然函数具有较低的弯曲度（图形轮廓相距很远），因此许多数据集的参数将产生一个合理的模型。从式 (2-46) 可知，低弯曲度将对应于高 $\hat{\mathbf{w}}$ 的协方差。费舍尔信息矩阵和协方差矩阵为：

$$\mathcal{I} = \begin{bmatrix} 50.0000 & 24.3311 \\ 24.3311 & 15.8953 \end{bmatrix}, \text{cov}\{\hat{\mathbf{w}}\} = \begin{bmatrix} 0.0784 & -0.1200 \\ -0.1200 & 0.2466 \end{bmatrix}$$

很难知道在没有上下文的情况下，这些是否对应于高或低的信息量或协方差。这一点可以通过将它们与第二个数据集（图 2-15 中的第二行）获得的情况进行比较得出。该数据集的噪声较低且对应的似然曲线弯曲度很高（图形轮廓距离很近）。在这种情况下，信息矩阵和协方差矩阵为：

$$\mathcal{I} = \begin{bmatrix} 1.2500 \times 10^3 & 0.6083 \times 10^3 \\ 0.6083 \times 10^3 & 0.3974 \times 10^3 \end{bmatrix}, \text{cov}\{\hat{\mathbf{w}}\} = \begin{bmatrix} 0.0031 & -0.0048 \\ -0.0048 & 0.0099 \end{bmatrix}$$

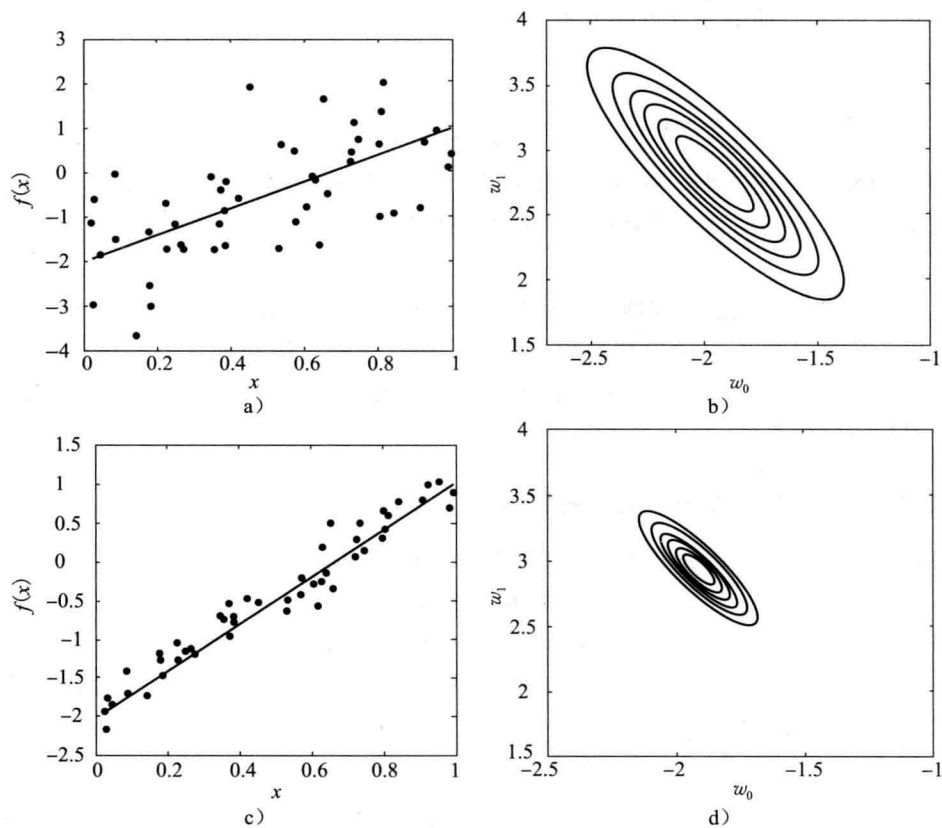


图 2-15 具有不同噪声程度和相应似然函数的两个数据集的例子

它明显具有较高的 \mathcal{I} 值和较低的 $\text{cov}\{\hat{\mathbf{w}}\}$ 值。

2.9.2 与实验数据比较

在 2.9 节的开始，我们根据式 (2-39) 的模型生成了对应一组输入的数据集。如果我们用 $\hat{\mathbf{w}}_s$ 来表示第 s 个数据集的参数，则实验数据的协方差矩阵可以通过如下公式计算：

$$\widehat{\text{cov}\{\hat{\mathbf{w}}\}} = \frac{1}{S} \sum_{s=1}^S (\hat{\mathbf{w}}_s - \hat{\boldsymbol{\mu}})(\hat{\mathbf{w}}_s - \hat{\boldsymbol{\mu}})^T$$

其中

$$\hat{\boldsymbol{\mu}} = \frac{1}{S} \sum_{s=1}^S \hat{\mathbf{w}}_s$$

使用图 2-13 中所用的值，实验协方差矩阵为：

$$\widehat{\text{cov}\{\hat{\mathbf{w}}\}} = \begin{bmatrix} 0.0627 & -0.0809 \\ -0.0809 & 0.1301 \end{bmatrix}$$

利用式 (2-45) 及其真实值 $\sigma^2=0.5^2$ ，理论协方差矩阵为

$$\text{cov}\{\hat{\mathbf{w}}\} = \begin{bmatrix} 0.0638 & -0.0821 \\ -0.0821 & 0.1317 \end{bmatrix}$$

它非常接近实验数据。通常，我们没有深不可测的数据，因此我们可以用理论协方差矩阵来帮助我们理解数据中的变异性。非对角元素是负值——即增大其中一个参数将引起其他参数减小。

为了计算理论协方差矩阵，使用了真实噪声方差。如果我们考虑任意的数据集，那么我们（使用式 (2-35)）能够估计方差为 $\sigma^2 = 0.2080$ （真实值是 $\sigma^2 = 0.25$ ），使用估计方差的协方差矩阵是

$$\text{cov}\{\hat{\mathbf{w}}\} = \begin{bmatrix} 0.0530 & -0.0683 \\ -0.0683 & 0.1095 \end{bmatrix}$$

由于 σ^2 的估计值比真实值小，所以该矩阵的值也比使用真实噪声值的矩阵值小。这表明不确定性被低估了，我们的预测过度自信。最大似然噪声的低估问题将在 2.10.2 节做更全面的讨论。

在 2.9 节的开始，我们发现准确的噪声值改变也改变了参数的估计值。在现实中，我们不能生成用于估计参数值不确定性的数据集。然而，我们已经得到了一个 $\hat{\mathbf{w}}$ 的协方差表达式，能用于评价参数的不确定性。在转向预测的变异性前，我们将关注奥运会数据的最大似然值所存在的不确定性。

2.9.3 模型参数的变异性——奥运会数据

现在，使用相似的奥运会男子 100 米数据和标准线性函数

$$f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \mathbf{x}$$

我们知道， \mathbf{w} 的最大似然值 $\hat{\mathbf{w}}$ 是 $[36.4165, -0.0133]^T$ （来源于式 (2-33)）。最大似然值的方差 $\hat{\sigma}^2$ 可以用式 (2-37) 计算， $\hat{\sigma}^2 = 0.0503$ 。使用式 (2-45)，用 $\hat{\sigma}^2$ 表示 σ^2 的估计值，我们能够计算估计值的协方差矩阵：

$$\text{cov}\{\hat{\mathbf{w}}\} = \begin{bmatrix} 5.7972 & -0.0030 \\ -0.0030 & 1.5204 \times 10^{-6} \end{bmatrix}$$

考虑对角元素，我们可以发现 $\hat{\mathbf{w}}_0 (5.7972)$ 的方差比 $\hat{\mathbf{w}}_1 (1.5204 \times 10^{-6})$ 的方差大，这表明我们能够忍受 $\hat{\mathbf{w}}_0$ 具有比 $\hat{\mathbf{w}}_1$ 大的变化，且仍然维持一个较合理的好模型。在某种程度上，这可以用这样一个事实来解释， $\hat{\mathbf{w}}_0$ 有更大的绝对值。非对角元素的负值性告诉我们，如果我们稍微增加 $\hat{\mathbf{w}}_0$ 或 $\hat{\mathbf{w}}_1$ 的值，那么势必引起其他值的稍微减小。这是比较直观的——如果我们稍微增大 $\hat{\mathbf{w}}_0$ ，那么整个直线将上移， $\hat{\mathbf{w}}_1$ 的最好值将会稍微减小（从而产生一个更加陡峭的负梯度）以更加接近所有的数据点。

了解 $\text{cov}(\hat{\mathbf{w}})$ 含义的另一种方法是关注模型的变异性。为此，我们可以假设 $\hat{\mathbf{w}}$ 是高斯分布的随机变量

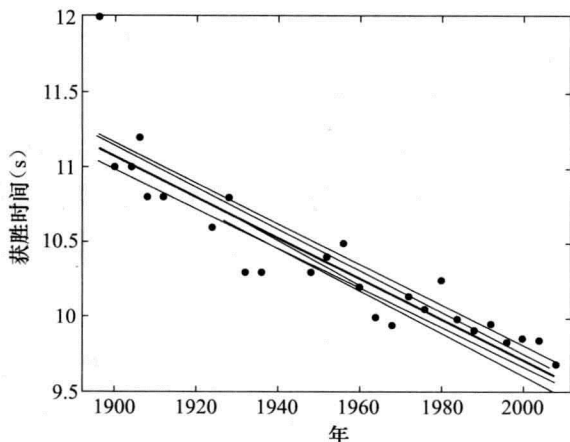


图 2-16 式 (2-48) 所示分布的 \mathbf{w} 的 10 个样本

$$\boldsymbol{w} \sim \mathcal{N}(\hat{\boldsymbol{w}}, \text{cov}\{\hat{\boldsymbol{w}}\}) \quad (2-48)$$

从这个密度，我们可以抽样很多 \boldsymbol{w} 示例来勾画模型。一个10个样本的例子如图2-16所示，我们可以发现在10个样本中只有 w_1 的坡度有很小的改变，但是如果我们将回推至第0年，这个小小的改变有可能引起 w_0 较大的变化。这可以通过已经讨论过的 $\text{cov}(\hat{\boldsymbol{w}})$ 来反映。使用模型参数的一种分布而不是其最优值的思想在机器学习中非常重要，第3章将进行介绍。

2.10 预测值的变异性

在第1章，我们做了一些关于未来奥运会100米获胜时间的预测。我们认为这些预测不会很有用，因为它们是以非常精确的形式呈现。预测出一个我们认为取胜时间可能分布范围的值应该更加合理。如果我们非常确定我们的预测值，那么这个范围可能比较小；如果我们不太确定，范围可能就比较大了。因此，正如我们得到参数估计值 $\hat{\boldsymbol{w}}$ 的变异范围一样，得到预测值的变异范围或者不确定性也就比较有意义。假设我们观察了一组新的属性 $\boldsymbol{x}_{\text{new}}$ ，我们将要预测新的输出 t_{new} 以及其相应的变异度 σ_{new}^2 。

为了预测 t_{new} ，用将 $\boldsymbol{x}_{\text{new}}$ 乘以最优模型参数 $\hat{\boldsymbol{w}}$

$$t_{\text{new}} = \hat{\boldsymbol{w}}^T \boldsymbol{x}_{\text{new}} \quad (2-49)$$

为了证明这样做是有意义的，我们可以计算其期望值：

$$E_{p(t|\boldsymbol{x}, \boldsymbol{w}, \sigma^2)}\{t_{\text{new}}\} = E_{p(t|\boldsymbol{x}, \boldsymbol{w}, \sigma^2)}\{\hat{\boldsymbol{w}}\}^T \boldsymbol{x}_{\text{new}} = \boldsymbol{w}^T \boldsymbol{x}_{\text{new}}$$

这里我们使用了式(2-40)。预测值的期望值就是新的输入值乘以真实的 \boldsymbol{w} 。在2.28节，我们得到了一个更通用方差表达式。在我们的情况中，就是

$$\sigma_{\text{new}}^2 = \text{var}\{t_{\text{new}}\} = E_{p(t|\boldsymbol{x}, \boldsymbol{w}, \sigma^2)}\{t_{\text{new}}^2\} - (E_{p(t|\boldsymbol{x}, \boldsymbol{w}, \sigma^2)}\{t_{\text{new}}\})^2$$

为了评估该表达式，我们首先需要将 $t_{\text{new}} = \hat{\boldsymbol{w}}^T \boldsymbol{x}_{\text{new}}$ 代入：

$$\begin{aligned} \text{var}\{t_{\text{new}}\} &= E_{p(t|\boldsymbol{x}, \boldsymbol{w}, \sigma^2)}\{(\hat{\boldsymbol{w}}^T \boldsymbol{x}_{\text{new}})^2\} - (\boldsymbol{w}^T \boldsymbol{x}_{\text{new}})^2 \\ &= E_{p(t|\boldsymbol{x}, \boldsymbol{w}, \sigma^2)}\{\boldsymbol{x}_{\text{new}}^T \hat{\boldsymbol{w}} \hat{\boldsymbol{w}}^T \boldsymbol{x}_{\text{new}}\} - \boldsymbol{x}_{\text{new}}^T \boldsymbol{w} \boldsymbol{w}^T \boldsymbol{x}_{\text{new}} \end{aligned}$$

代入类似 $\hat{\boldsymbol{w}}$ 的表达式：

$$\text{var}\{t_{\text{new}}\} = \boldsymbol{x}_{\text{new}}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T E_{p(t|\boldsymbol{x}, \boldsymbol{w}, \sigma^2)}\{t\} \boldsymbol{X} (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{x}_{\text{new}} - \boldsymbol{x}_{\text{new}}^T \boldsymbol{w} \boldsymbol{w}^T \boldsymbol{x}_{\text{new}}$$

使用 $\text{cov}\{t\}$ 的表达式(式(2-43))，可以计算期望值并简化表达式：

$$\begin{aligned} \text{var}(t_{\text{new}}) &= \boldsymbol{x}_{\text{new}}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T (\sigma^2 \boldsymbol{I} + \boldsymbol{X} \boldsymbol{w} \boldsymbol{w}^T \boldsymbol{X}^T) \boldsymbol{X} (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{x}_{\text{new}} - \boldsymbol{x}_{\text{new}}^T \boldsymbol{w} \boldsymbol{w}^T \boldsymbol{x}_{\text{new}} \\ &= \sigma^2 \boldsymbol{x}_{\text{new}}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{x}_{\text{new}} + \boldsymbol{x}_{\text{new}}^T \boldsymbol{w} \boldsymbol{w}^T \boldsymbol{x}_{\text{new}} - \boldsymbol{x}_{\text{new}}^T \boldsymbol{w} \boldsymbol{w}^T \boldsymbol{x}_{\text{new}} \\ &= \sigma^2 \boldsymbol{x}_{\text{new}}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{x}_{\text{new}} \end{aligned}$$

注意，如果将 $\text{cov}(\hat{\boldsymbol{w}})$ (式(2-41))代入该表达式，则该表达式可以改写成：

$$\sigma_{\text{new}}^2 = \boldsymbol{x}_{\text{new}}^T \text{COV}\{\hat{\boldsymbol{w}}\} \boldsymbol{x}_{\text{new}}$$

总之，预测值和相应的方差为：

$$t_{\text{new}} = \boldsymbol{x}_{\text{new}}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{t} = \boldsymbol{x}_{\text{new}}^T \hat{\boldsymbol{w}} \quad (2-50)$$

$$\sigma_{\text{new}}^2 = \sigma^2 \boldsymbol{x}_{\text{new}}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{x}_{\text{new}} \quad (2-51)$$

σ^2 是数据噪声的真实方差。在此处，我们可以用估计值 $\hat{\sigma}^2$ 替代它。

2.10.1 预测值的变异性——一个例子

图2-17a显示了方程 $f(x) = 5x^3 - x^2 + x$ 和取样点以及被均值为0、方差为1000的高斯分布的噪声干扰的情况。在图2-17b、c、d中，我们可以发现 $t_{\text{new}} \pm \sigma_{\text{new}}^2$ 分别为线性、立方和6阶模型(MATLAB脚本: predictive_variance_example.m)。

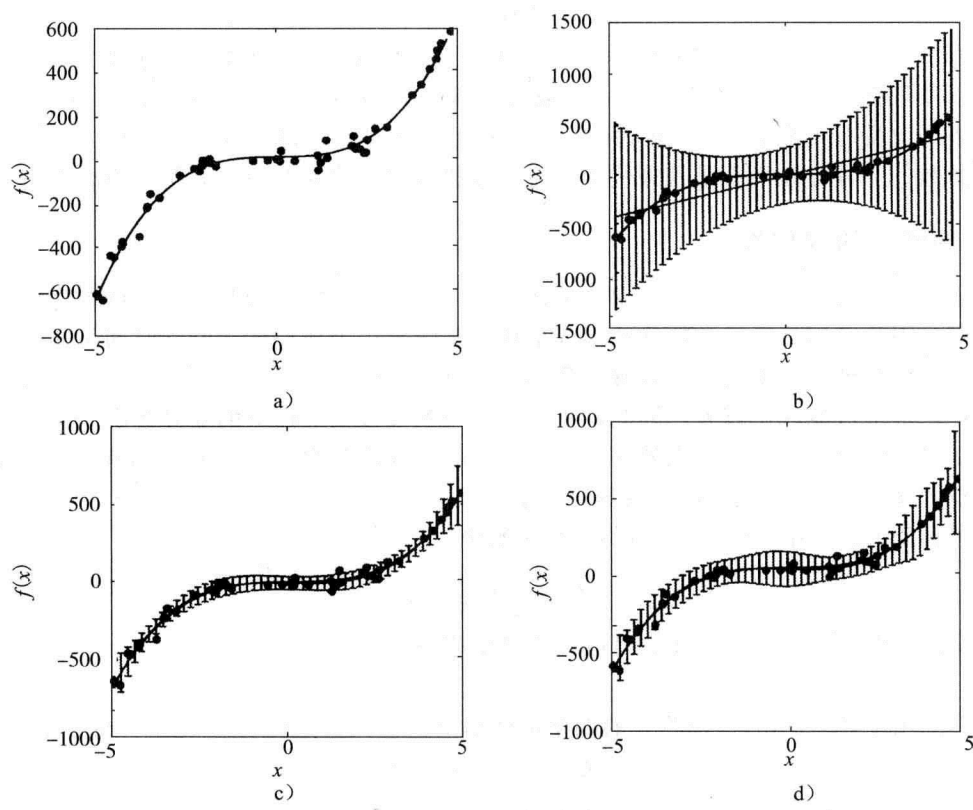


图 2-17 a) 事例数据集；b)、c)、d) 分别为线性、立方和 6 阶模型的预测误差图

线性模型具有非常高的预测方差。它不能非常好地对数据的真实趋势建模，并且数据的很多的变异性被假定为噪声。立方模型能够更好地对这些趋势建模（它是正确的阶数），并且它反映在了它更加可信的预测值中。6 阶模型过度复杂——它具有太大的随意度，因此能很好地拟合较大变化范围的参数值。 $\hat{\mathbf{w}}$ 的这种不确定性通过增加预测的变异性来实现——如果我们不能确定参数的估计值，那么我们也不能确定它的预测值。这一点可以通过计算 3 阶和 6 阶模型的协方差矩阵 $\text{cov}\{\hat{\mathbf{w}}\}$ ，然后像 2.93 节那样取样来证实。图 2-18 显示了 3 阶和 6 阶模型的 $\hat{\mathbf{w}}$ 和 $\text{cov}\{\hat{\mathbf{w}}\}$ 高斯分布的函数图（将图缩小为一个小 x 范围内且黑线表示真实函数）（MATLAB 脚本：predictive_variance_example.m）。在 6 阶模型中，清楚地显示了可能函数的不确定性随着参数不确定性的增长而增大的趋势。

85

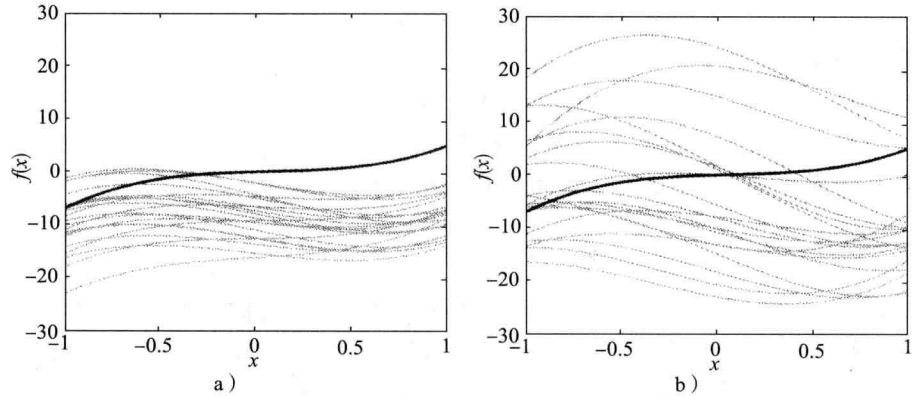


图 2-18 图 2-17a 中显示数据集的参数函数，具有均值 $\hat{\mathbf{w}}$ 和协方差 $\text{cov}\{\hat{\mathbf{w}}\}$ 的高斯分布

最后有趣的一点是，对于所有的模型，预测值的方差都随着我们靠近数据的边缘而增大。该模型在具有较少数据的领域里具有较低的可信度——一个吸引人的特点。第1章指出，对无限期未来的预测（即超出训练数据的范围）是毫无意义的。现在，我们有了一个能够预测超出训练数据范围的模型，但是这样做将会增加不确定性，这似乎更有用。我们还注意到它影响数据的中间部分（尤其是图2-17d），其有一个小的间隙（在 $x=1$ 周围没有太多的数据）。练习EX 2.12将有机会更进一步地研究这种影响。

2.10.2 估计值的期望值

在2.9.1节中，我们计算了估计值 $\hat{\mathbf{w}}$ 的期望值。该期望值用来生成 $p(t|\mathbf{X}, \mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2 \mathbf{I})$ 的密度，这里面计算一次：

$$\begin{aligned} E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\hat{\mathbf{w}}\} &= E_{p(t|\mathbf{X})}\{(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}\} \\ &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T E_{p(t|\mathbf{X})}\{\mathbf{t}\} \\ &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X} \mathbf{w} \\ &= \mathbf{I} \mathbf{w} = \mathbf{w} \end{aligned}$$

其中我们使用了 $\hat{\mathbf{w}}$ 的表达式（ $\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}$ ），并且事实是高斯随机变量（ \mathbf{t} ）的期望值等于高斯（ $\mathbf{X}\mathbf{w}$ ）的均值。因此，估计值 $\hat{\mathbf{w}}$ 的期望值是真实值 \mathbf{w} 。这是 $\hat{\mathbf{w}}$ 的一个很重要的属性，它告诉我们 $\hat{\mathbf{w}}$ 是一个无偏估计值——它不会一直太高或太低。另一种考虑这点的方式是2.9节开始部分的实验。对于一组特征值 $x_1, x_2, x_3, \dots, x_N$ ，我们产生了一组响应值，并关注不同噪声对 $\hat{\mathbf{w}}$ 有多大的影响。由于 $\hat{\mathbf{w}}$ 是无偏的，所以平均来讲，它应该是正确的。因此，如果我们采用实验中所有 $\hat{\mathbf{w}}$ 的平均值，那么将会非常接近真实值。事实上，我们采用 $\hat{\mathbf{w}}_0 = -2.0007$ 和 $\hat{\mathbf{w}}_1 = 3.0008$ 的平均值，它们都非常接近真实值： $w_0 = -2$ 和 $w_1 = 3$ 。

我们对噪声方差的估计值 σ^2 可以采用同样的处理，式(2-37)中 $\hat{\sigma}^2$ 的表达式为：

$$\hat{\sigma}^2 = \frac{1}{N} (\mathbf{t}^T \mathbf{t} - \mathbf{t}^T \mathbf{X} \hat{\mathbf{w}})$$

采用 $p(t|\mathbf{X}, \mathbf{w}, \sigma^2)$ 的期望值并进行一些操作，可以得到：

$$\begin{aligned} E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\hat{\sigma}^2\} &= \frac{1}{N} E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{t}^T \mathbf{t} - \mathbf{t}^T \mathbf{X} \hat{\mathbf{w}}\} \\ &= \frac{1}{N} E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{t}^T \mathbf{t} - \mathbf{t}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}\} \\ &= \frac{1}{N} E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{t}^T \mathbf{t}\} \\ &\quad - \frac{1}{N} E_{p(t|\mathbf{X}, \mathbf{w}, \sigma^2)}\{\mathbf{t}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}\} \end{aligned} \quad (2-52)$$

前面我们已经看到了形如 $\mathbf{t} \mathbf{t}^T$ 的表达式，但不是 $\mathbf{t}^T \mathbf{t} (= \mathbf{t}^T \mathbf{I} \mathbf{t})$ 或者 $\mathbf{t}^T \mathbf{A} \mathbf{t}$ 。当 \mathbf{t} 是高斯随机变量时，表达式 $\mathbf{t}^T \mathbf{A} \mathbf{t}$ 的期望是：

$$\begin{aligned} \mathbf{t} &\sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ E_{p(\mathbf{t})}\{\mathbf{t}^T \mathbf{A} \mathbf{t}\} &= \text{Tr}(\mathbf{A} \boldsymbol{\Sigma}) + \boldsymbol{\mu}^T \mathbf{A} \boldsymbol{\mu} \end{aligned}$$

其中 $\text{Tr}()$ 是迹函数(参见注解2.8)。式(2-52)右边的第一项 $\mathbf{A} = \mathbf{I}_N$ （注意 $\mathbf{t}^T \mathbf{t} = \mathbf{t}^T \mathbf{I}_N \mathbf{t}$ ，其中 \mathbf{I}_N 是一个 $N \times N$ 的单位矩阵）；第二项， $\mathbf{A} = \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ 。在 $\boldsymbol{\mu} = \mathbf{X} \mathbf{w}$ 和 $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}_N$ 在两种情况下，将必要的值代入式(2-52)可得：

注解2.8 (阵迹)：方阵 \mathbf{A} 的阵迹，表示为 $\text{Tr}(\mathbf{A})$ 是矩阵 \mathbf{A} 的对角元素的和，如果

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1D} \\ A_{21} & A_{22} & \cdots & A_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ A_{D1} & A_{D2} & \cdots & A_{DD} \end{bmatrix}$$

则

$$\text{Tr}(\mathbf{A}) = \sum_{d=1}^D A_{dd}$$

随后, 若 $\mathbf{A} = \mathbf{I}_D$, 即 $D \times D$ 单位矩阵,

$$\text{Tr}(\mathbf{I}_D) = \sum_{d=1}^D 1 = D$$

一个将经常使用的单位矩阵是

$$\text{Tr}(\mathbf{AB}) = \text{Tr}(\mathbf{BA})$$

同样, 一个标准矩阵的阵迹等于其标准值 (一个标准矩阵是形如 1×1 的矩阵), 即

$$\text{Tr}(a) = a$$

或者, 如果 $\mathbf{w} = [w_1, \dots, w_D]^T$

$$\text{Tr}(\mathbf{w}^T \mathbf{w}) = \mathbf{w}^T \mathbf{w}$$

因为 $\mathbf{w}^T \mathbf{w}$ 的结果为一个标准矩阵。

$$\begin{aligned} E_{p(t|x, \mathbf{w}, \sigma^2)} \{\hat{\sigma}^2\} &= \frac{1}{N} (\text{Tr}(\sigma^2 \mathbf{I}_N) + \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w}) \\ &\quad - \frac{1}{N} (\text{Tr}(\sigma^2 \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T) + \mathbf{w}^T \mathbf{X}^T \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X} \mathbf{w}) \end{aligned}$$

因为 $\mathbf{I}_N \mathbf{I}_N = \mathbf{I}_N$ 。根据定义, 现在 $\text{Tr}(\sigma^2 \mathbf{A}) = \sigma^2 \text{Tr}(\mathbf{A})$, $\text{Tr}(\mathbf{I}_N) = N$ 。这样可以简化表达式为:

$$\begin{aligned} E_{p(t|x, \mathbf{w}, \sigma^2)} \{\hat{\sigma}^2\} &= \sigma^2 + \frac{1}{N} \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} - \frac{\sigma^2}{N} \text{Tr}(\mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T) - \frac{1}{N} \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} \\ &= \sigma^2 - \frac{\sigma^2}{N} \text{Tr}(\mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T) \\ &= \sigma^2 \left(1 - \frac{1}{N} \text{Tr}(\mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T) \right) \end{aligned}$$

最后, 我们需要利用 $\text{Tr}(\mathbf{AB}) = \text{Tr}(\mathbf{BA})$ 的事实, 因此迹函数中的第一个 \mathbf{X} 可以移动到最后:

$$\begin{aligned} E_{p(t|x, \mathbf{w}, \sigma^2)} \{\hat{\sigma}^2\} &= \sigma^2 \left(1 - \frac{1}{N} \text{Tr}((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X}) \right) \\ &= \sigma^2 \left(1 - \frac{1}{N} \text{Tr}(\mathbf{I}_D) \right) \\ &= \sigma^2 \left(1 - \frac{D}{N} \right) \end{aligned} \quad (2-53)$$

其中 D 是特征的个数 (\mathbf{X} 中的列数)。

假设 $D < N$ (即我们测量的每一个数据点的特征数小于数据点的个数), 那么平均方差的估计值平均比实际方差小:

$$E_{p(t|x, \mathbf{w}, \sigma^2)} \{\hat{\sigma}^2\} < \sigma^2$$

与 $\hat{\mathbf{w}}$ 不同, 该估计值是有偏的。

我们可以返回到前面虚构的实验来观察这种偏差。所有数据集的 $\hat{\sigma}^2$ 的平均值为 0.2264。真实值 $\hat{\sigma}^2 = 0.5^2 = 0.25$ 。我们可以发现, 平均值确实太小了。对于这个例子, $D = 2$, $N = 20$, 因此我们理论上可以接受的期望值是

$$E_{p(t|x, \mathbf{w}, \sigma^2)} \{\hat{\sigma}^2\} = \sigma^2 \left(1 - \frac{D}{N} \right) = 0.25 \left(1 - \frac{2}{20} \right) = 0.2250$$

其接近于观察到的平均值。

从式 (2-53)，我们注意到降低这种偏置的一种方式使 D/N 变小。 D 一般是固定的，但我们可以增大 N 。在图 2-19 中，我们可以看到将 N 从 20 增加到 10 000 的影响 (MATLAB 脚本: w_variatiion_demo.m)。随着数据点的增加，理论曲线 (虚线) 及实验曲线 (实线) (通过使用不同的观测值, N , 重复前面的实验) 非常接近且涵盖了真实值 $\sigma^2 = 0.25$ 。

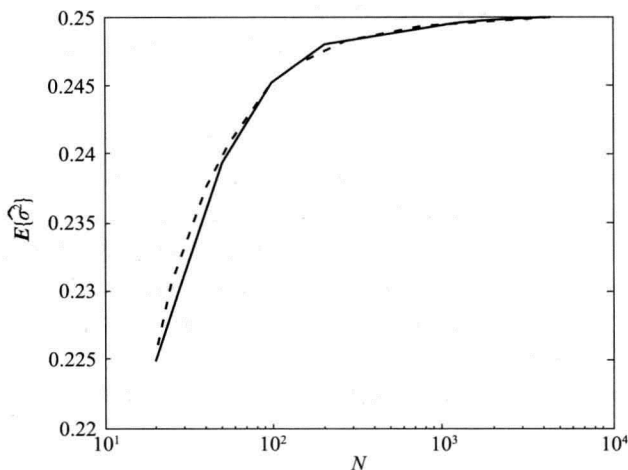


图 2-19 随着数据点的增加, $E_{p(t|X, w, \sigma^2)} \{\hat{\sigma}^2\}$ 的理论与实验估计值的变化

它可以提供一个关于 $\hat{\sigma}^2$ 偏差的直观解释。 σ^2 的最大似然值估计值的表达式是:

$$\hat{\sigma}^2 = \frac{1}{N} (t^T t - t^T X \hat{w}) \quad (2-54)$$

它可以改写为与预测值和真实值之间的均方误差和相同的表达式 (参见练习 EX 2.11):

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N (t_n - x_n^T \hat{w})^2$$

89

这就告诉我们, 模型越接近真实数据, $\hat{\sigma}^2$ 就越小。现在, 想象 w 的真实值与估计值 \hat{w} , 哪个更接近真实数据? 最大似然值估计值 \hat{w} 等于最小损失估计。根据定义, 它们是一组接近真实数据的参数, 因此可以最小化 $\hat{\sigma}^2$ 。如果式 (2-54) 中使用 w 的真实值代替 \hat{w} , 那么得到的 $\hat{\sigma}^2$ 的值将会大于或者等于使用 \hat{w} 得到的值。因为我们将发现最小化噪声 w 的平均值, 所以将随着噪声程度低于真实值而中止。

2.10.3 小结

在前面的几节中, 我们已经涵盖了很多材料。关于随机变量的介绍提供了用于对真实数据与我们提出的决定性模型之间误差建模的最基本理论。通过显式地对这些误差建模, 我们已经发现, 如果将数据噪声假设为一种正态分布, 那么第 1 章所说的最小二乘解将如何与最大似然值相同。使用似然法的优点是能够定量参数估计的不确定性, 进而估计预测的不确定性。这就让我们能够从准确的预测值 (这一定是错误的) 转变成一定范围的值 (如 $t_{\text{new}} \pm \sigma^2$)。在很多应用中, 这将更加有用。最后, 我们关注最大似然参数的一些理论特性, 并发现尽管我们的估计值 \hat{w} 是无偏的, 但平均来看, $\hat{\sigma}^2$ 是有偏的, 非常低。

2.11 练习

EX 2.1 100 米线性回归 (见图 2-1) 的误差模型会是离散或连续随机变量的最佳模型吗?

EX 2.2 如果遵循这样的事实，当掷骰子时，所有结果具有相同的概率，使用式 (2-1) 与式 (2-2) 所给的限制条件，计算骰子 6 个面每一面朝上的概率。

EX 2.3 Y 是一个能取任意正整数的随机变量，其结果的似然值用泊松概率密度函数给出

$$p(y) = \frac{\lambda^y}{y!} \exp\{-\lambda\}$$

如果遵循这样的事实，对于一个离散随机变量，概率密度函数给出了每个事件的概率且这些概率是可加性的。

(a) 计算 $\lambda=5$ 时， $Y \leq 4$ 的概率，即 $P(Y \leq 4)$ 。

(b) 利用 (a) 的结果且有一个结果已经发生，计算 $Y > 4$ 的概率（提示， $Y \leq 4$ 和 $Y > 4$ 中有一个事件已经发生）。

EX 2.4 Y 是一个正态密度的随机变量， $p(y) = U(a, b)$ ，得到 $E_{p(y)}\{\sin(y)\}$ 。注意， $\int \sin(y) dy = -\cos(y)$ 。

90

当 $a=0$ ， $b=1$ 时，计算 $E_{p(y)}\{\sin(y)\}$ 。修改 `approx_expexted_value.m` 来计算该值基于抽样的近似值并观察该近似值如何随着抽样数量的改变而改变。

EX 2.5 假设 $p(\mathbf{w})$ 是式 (2-28) 所示的 D 维向量 \mathbf{w} 的高斯密度概率。通过展开向量定义公式并重写，证明 $\Sigma = \sigma^2 \mathbf{I}$ 是协方差矩阵，假设 \mathbf{w} 的 D 个元素之间是独立的。你需要注意的是，只有对角元素矩阵 ($\sigma^2 \mathbf{I}$) 的决定性因素是其对角值与通过颠倒对角元素而构建的矩阵的反向乘积（提示，指数的乘积可以表示成指数的和）。

EX 2.6 利用与 EX 2.5 相同的设置，看看如果我们使用对角线上具有不同元素的对角协方差矩阵，将会发生什么，即

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_D^2 \end{bmatrix}$$

EX 2.7 证明对于 1 阶多项式，对数似然值的 2 阶导数的 Hessian 矩阵等于式 (1-9) 的 2 阶导数（它们的不同只是乘以不同的常数）。

EX 2.8 假设具有 N 个值的数据集 $x_1, x_2, x_3, \dots, x_N$ 来源于高斯分布的抽样。假设数据是 IID，找出高斯均值和方差的最大似然估计（提示，从写出 N 个数据点的联合似然开始，并注意指数函数的乘积可以写成指数函数的指数和形式）。

EX 2.9 假设一个具有 N 个二元值数据集 $x_1, x_2, x_3, \dots, x_N$ 来源于伯努利分布的抽样。计算伯努利参数的最大似然估计。

EX 2.10 求具有 N 个观测值 $x_1, x_2, x_3, \dots, x_N$ 的多元高斯密度的均值和协方差矩阵的最大似然估计。

EX 2.11 证明在线性模型中，噪声方差的最大似然估计

$$\hat{\sigma}^2 = \frac{1}{N} (\mathbf{t}^T \mathbf{t} - \mathbf{t}^T \mathbf{X} \hat{\mathbf{w}})$$

可以写成

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N (t_n - \mathbf{x}_n^T \hat{\mathbf{w}})^2$$

（提示：从第二个表达式反向计算）

EX 2.12 使用 `predictive_variance_example.m` 生成一个数据集，并删除所有 $-1.5 \leq x \leq 1.5$ 的值。分析这样做对预测值方差范围的影响。

91

EX 2.13 计算伯努利分布参数的费舍尔信息矩阵。

92

EX 2.14 计算多元高斯密度中均值矩阵的费舍尔信息矩阵。

其他阅读材料

[1] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

这本书是包含许多机器学习概念的好资源。尤其是，它包括了偏差-方差权衡的详细讨论。

- [2] J.H. McColl. *Probability*. Elsevier, 1995.

关于概率论的一个非常简洁的介绍。

- [3] Paul Meyer. *Introductory Probability and Statistical Applications*. Addison-Wesley, 1978.

介绍概率论的一个极好的资源。

- [4] J. Rosenthal. *A First Look at Rigorous Probability Theory*. World Scientific Publishing Company, 2006.

对于开始探索测度理论——概率论的一个分支，这是一本十分好的书。

- [5] Michael Tipping and Christopher Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(3):611–622, 1999.

最大似然值的一个有趣应用。这里，将它应用到概率方法中，第一概率方法是主成分分析的经典统计问题。

机器学习的贝叶斯方法

在前面的章节中，讲到显式地在模型中增加噪声允许我们不只是完成点预测。特别是能够量化参数估计和后续预测过程中的不确定性。一旦在参数估计中考虑不确定性，将参数本身作为随机变量只是一个小的步骤。贝叶斯方法在机器学习中越来越重要，接下来的两章将介绍许多人在这一领域发现的挑战问题。本章首先通过两个例子介绍贝叶斯统计的一些基本思想，只是贝叶斯推理所需的计算往往不能方便地解析处理。在第4章中，将会介绍机器学习领域中流行的三个近似方法。

3.1 硬币游戏

设想你漫步在露天市场上，经过一个货摊，这里的顾客正在参与一个投币游戏。货摊主为每个顾客投10次硬币。如果硬币正面朝上的次数等于或者小于6次，则该顾客收益自己押注的1镑和额外收益的1镑。如果硬币正面朝上的次数大于6次，则货摊主收益顾客押注的1美元。二项分布（见2.3.2节）描述了在 N 次二值事件中出现一定次数成功（正面朝上）的概率。假设每次投硬币正面朝上的概率为 r ，则投 N 次硬币有 y 次正面朝上的概率为：

$$P(Y = y) = \binom{N}{y} r^y (1-r)^{N-y} \quad (3-1)$$

假设硬币是公平的，因此设 $r=0.5$ 。当 $N=10$ 时， Y 的概率分布函数如图3-1所示，其中阴影部分对应 $y \leq 6$ 。利用式(3-1)可以计算出赢得游戏的概率，也就是 Y 小于或者等于6的概率 $P(Y \leq 6)$ ：

$$\begin{aligned} P(Y \leq 6) &= 1 - P(Y > 6) = 1 - [P(Y = 7) + P(Y = 8) + P(Y = 9) + P(Y = 10)] \\ &= 1 - [0.1172 + 0.0439 + 0.0098 + 0.0010] \\ &= 0.8281 \end{aligned}$$

这看起来是一个很好的游戏，你将以0.8281的概率使你的钱翻倍增长。你也可以计算从这个游戏中获益的期望值。随机变量 X 的函数 $f(X)$ 的期望值可以如下计算（见2.2.8节）：

$$E_{P(x)}\{f(X)\} = \sum_x f(x)P(x)$$

其中要在随机变量的所有可能取值上计算和。设 X 是一个随机变量，如果我们（顾客）赢了，则 X 取值为1；否则，取值为0。显然， $P(X=1)=P(Y \leq 6)$ 。如果我们赢了（ $X=1$ ），我们获得2镑（我们自己的1镑加上额外收益的1镑），因此 $f(1)=2$ 。如果我们输了，我们什么也没有获得，因此 $f(0)=0$ 。我们收益的期望值为：

$$f(1)P(X=1) + f(0)P(X=0) = 2 \times P(Y \leq 6) + 0 \times P(Y > 6) = 1.6562$$

如果你花了1镑参与游戏，平均情况下你每场游戏获得 $(1.6562 - 1)$ 镑，差不多66便士。如果你玩了100次，则你可以获得65.62镑的收益。

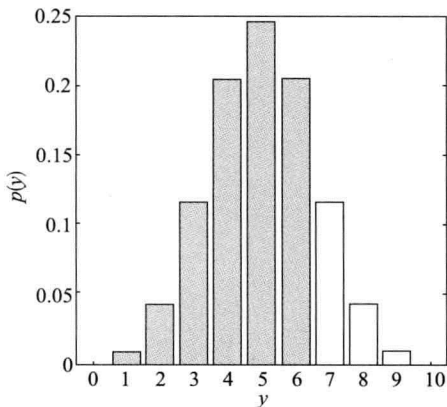


图3-1 当 $N=10$ 、 $r=0.5$ 时的二项密度函数（式(3-1)）

给了你这样的机会，似乎应该要玩的。然而，等一会你就会注意到货摊主看起来很有理由富有并且只有少数顾客看起来能够获胜。也许计算的假设是错误的。之前我们的假设包括：

1) 正面朝上的次数是一个符合二项分布的随机变量，并且投一次硬币正面朝上的概率为 r 。

2) 硬币是公平的，也就是正面朝上的概率等于背面朝上的概率，即 $r=0.5$ 。看起来很难否定二项分布，因为事件确实只有两种可能的结果并且每次投币都是独立的。那么就只剩下 r （正面朝上的概率）了。我们假设硬币是公平的，即正面朝上和背面朝上的概率是相同的。也许不是这种情况呢？为了对此进行研究，我们把 r 当成一个参数（像前面章节的 ω 和 σ^2 ）拟合到某些数据上。

3.1.1 计算正面朝上的次数

假设在玩游戏的队伍里有 3 个人。第一个人玩游戏并且获得如下的正面和背面序列

H, T, H, H, H, H, H, H, H, H

9 个正面朝上、1 个背面朝上。这时我们可以按如下方式计算 r 的最大似然值。二项分布的似然值定义为：

$$P(Y = y | r, N) = \binom{N}{y} r^y (1-r)^{N-y} \quad (3-2)$$

取自然对数得：

$$L = \log P(Y = y | r, N) = \log \binom{N}{y} + y \log r + (N-y) \log(1-r)$$

像第 2 章一样，对这个表达式求微分，令它等于 0，求参数的最大似然值估计值：

$$\begin{aligned} \frac{\partial L}{\partial r} &= \frac{y}{r} - \frac{N-y}{1-r} = 0 \\ y(1-r) &= r(N-y) \\ y &= rN \\ r &= \frac{y}{N} \end{aligned}$$

将 $y=9$ 、 $N=10$ 代入，得 $r=0.9$ 。相应的分布函数如图 3-2 所示，重新计算顾客获胜的概率为 $P(Y \leq 6) = 0.0128$ 。这远远低于 $r=0.5$ 时的值。此时收益的期望值为：

$$2 \times P(Y \leq 6) + 0 \times P(Y > 6) = 0.0256$$

如果你花了 1 镑参与游戏，平均情况下每场游戏你获得 $0.0256 - 1 = -0.9744$ 镑，差不多损失 97 便士。 $P(Y \leq 6) = 0.0128$ 表明，每 100 个人里只有 1 个人能赢，但这似乎没有反映出获胜的人数。虽然这次投硬币的情况表明 $r=0.9$ ，但它看起来有点极端了，因为有好几个人已经赢了。

3.1.2 贝叶斯方法

上一节中，我们计算 r 的值是基于 10 次投币。考虑到投币具有随机特性，如果我们观察多个投币序列，我们每次都可能获得不同的 r

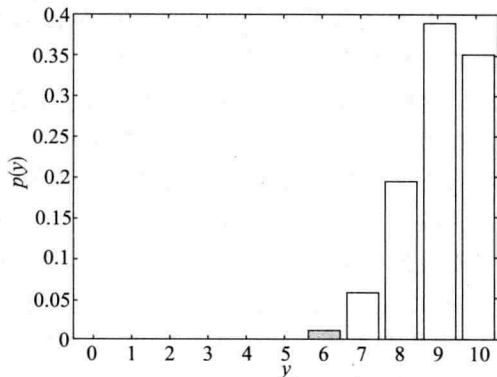


图 3-2 当 $N=10$ 、 $r=0.9$ 时的二项分布函数（式（3-1））

值。考虑这种情况, r 感觉有点像一个随机变量 R 。也许我们能学到一些关于 R 的分布而不是寻找一个特定的值。在前一节中, 我们看到通过计算获得的具体值严重受到短序列中特定投币的影响。不管这样的序列有多少, 我们发现 r 的值总是不确定的, 把 r 考虑为与一个与分布相关联的随机变量将有助于我们度量和理解这种不确定性。

尤其是, 定义随机变量 Y_N 为 N 次投币中正面朝上的次数, 我们能够计算 r 在 Y_N 上的条件分布:

$$p(r|y_N)$$

给定了该分布, 我们可以通过计算 $P(Y_{\text{new}} \leq 6|r)$ 关于 $p(r|y_N)$ 的期望值来获得顾客赢的期望概率:

$$P(Y_{\text{new}} \leq 6|y_N) = \int P(Y_{\text{new}} \leq 6|r)p(r|y_N)dr$$

其中 Y_{new} 是描述在未来 10 次投币中正面朝上次数的随机变量。

在 2.2.7 节中我们简单介绍了贝叶斯规则。贝叶斯规则允许我们颠倒两个随机变量的条件, 也就是从 $p(b|a)$ 计算 $p(a|b)$ 。这里我们感兴趣的是 $p(r|y_N)$, 如果我们颠倒条件, 就是 $p(y_N|r)$ ——在 N 次投币中正面朝上的次数, 其中一次投币中正面朝上的概率为 r 。这就是二项分布函数, 对任意的 y_N 和 r , 我们可以很容易计算。在上下文中, 贝叶斯规则为 (见式 (2-11)):

$$p(r|y_N) = \frac{P(y_N|r)p(r)}{P(y_N)} \quad (3-3)$$

这个公式在后面的章节中是非常重要的, 因此需要读者花点时间详细地理解这个公式。

似然值 $P(y_N|r)$: 在第 2 章中我们介绍了似然。这里它具有相同的意义: 对一个特定的 r 值 (我们的模型), 我们观察到数据 (在这种情况下, 数据是 y_N) 的可能性。如果 r 产生 y_N 的可能性大, 则似然值高; 否则, 似然值低。例如, 图 3-3 给出了两个不同场景下 r 与似然值 $P(y_N|r)$ 的函数关系。在第一个场景下, 数据包含 10 次投币 ($N=10$), 其中 6 次正面朝上。在第二个场景下, 投币次数为 100, 正面朝上的次数为 70。

图 3-3 揭示了似然值的两个重要属性。首先, 它不是概率密度。如果是概率密度, 则两个曲线下方的面积将等于 1。我们可以看到两个图的面积是完全不同的, 所有根本不会为 1。其次, 这两个场景告诉我们 r 值的范围是不同的。

在第一个场景下, r 较大范围 (大约为 $0.2 \leq r \leq 0.9$) 内似然值不为 0。第二个场景下, 似然值不为 0 的区域减少了很多 (大约为 $0.6 \leq r \leq 0.8$)。显然, 在第二个场景下, 我们有较多的数据 (投币次数为 100 而不是 10), 因此我们能够知道更多的 r 值。

先验分布 $p(r)$: 先验分布允许我们在没有看到任何数据之前表达我们认为 r 值是多少。为了说明这点, 我们考虑下面的 3 个实例。

- 1) 我们不知道任何关于投币和货摊主的信息。
- 2) 我们认为硬币 (因此货摊主) 是公平的。
- 3) 我们认为硬币 (因此货摊主) 倾向于正面朝上。

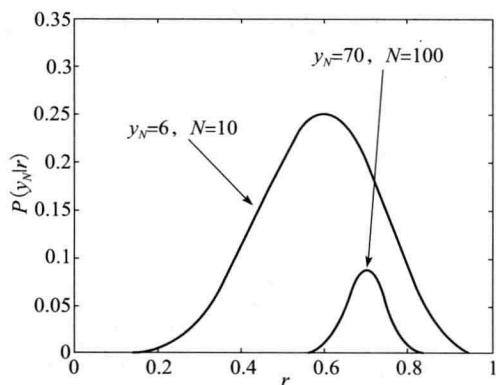


图 3-3 第二个场景中似然值 $p(y_N|r)$ 与 r 的函数关系

我们可以将这些信念编码为不同的先验分布。 r 可以在0和1之间任意取值,因此必须将模型化为一个连续随机变量。图3-4给出了三个密度函数,分别用于对三个不同的先验信念进行编码。

信念1表示为0和1之间的均匀密度,也就是对任意 r 值没有任何偏向。信念2给出了一个在 $r=0.5$ 时达到最大值的密度函数,该值表明我们期望硬币是公平的。该密度表明我们不希望 r 有太多的不同, r 值几乎都在0.4~0.6之间。我们投的大多数硬币都遵循信念2。最后,信念3认为硬币(货摊主)是有偏的。该密度表明 r 大于0.5,并且具有很大的不同。这是最好的,因为我们的信念是硬币是有偏的:在该阶段还没有认识到结果偏倚的程度。

在该阶段,我们不从3个场景中进行选择,因为看看这些不同信念对 $p(r|y_N)$ 的影响是很有趣的。

图3-4中三个函数的绘制并不是凭空捏造(子虚乌有)的。它们都是 β 概率密度函数的实例。 β 概率密度函数用于连续随机变量,取值范围在0和1之间,非常适合我们的实例。将具有参数 α 和 β 的随机变量 R 定义为:

$$p(r) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1} (1-r)^{\beta-1} \quad (3-4)$$

$\Gamma(a)$ 是 γ 函数(见2.5.2节)。在式(3-4)中, γ 函数保证密度是标准化的(也就是,它合并为1并且是一个概率密度函数)。尤其是,

$$\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} = \int_{r=0}^{r=1} r^{\alpha-1} (1-r)^{\beta-1} dr$$

保证了

$$\int_{r=0}^{r=1} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1} (1-r)^{\beta-1} dr = 1$$

参数 α 和 β 控制最终密度函数的形状,并且都是正的。图3-4显示的三个信念对应于如下的参数值:

- 1) 什么也不知道: $\alpha=1, \beta=1$ 。
- 2) 公平的硬币: $\alpha=50, \beta=50$ 。
- 3) 有偏的: $\alpha=5, \beta=1$ 。

这些值的选择问题是一个大问题。例如,为什么我们为有偏的硬币选择 $\alpha=5, \beta=1$ 呢?这不太容易回答。后面我们将看到beta分布可以解释为一些以前假设的硬币投掷。对于其他分布,类似情况是不可能的,我们将引入这样的概念:也许这些也应该看成是随机变量。与此同时,我们假设这些值是合理的。

y_N 的边缘分布 $P(y_N)$:我们公式里的第3个量 $P(y_N)$ 作为一个标准化的常量用于保证 $p(r|y_N)$ 是一个合理的密度。 y_N 的边缘分布是众所周知的,因为它通过联合密度 $p(y_N, r)$ 结合 r 计算出来的:

$$P(y_N) = \int_{r=0}^{r=1} p(y_N, r) dr$$

联合密度可以分解为:

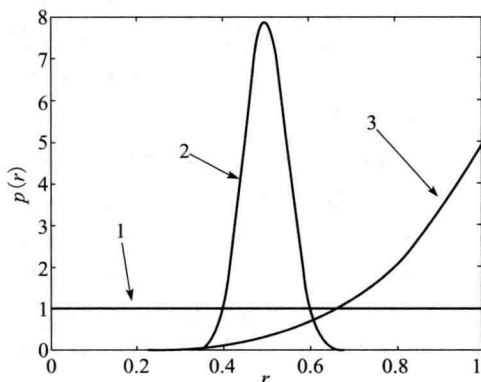


图3-4 三个不同场景对 r 的先验密度 $p(r)$ 示例

$$P(y_N) = \int_{r=0}^{r=1} P(y_N | r) p(r) dr$$

是先验和似然乘积在 r 的取值范围内的积分。

$P(y_N)$ 也称为**边缘似然值**，因为它是对所有参数值取平均后数据 y_N 的似然值。在 3.4.1 节中我们将看到它是模型选择中一个有意义的量，但不幸的是，除了极少数情况外，它是很难计算的。

后验分布 $p(r|y_N)$ ：后验是我们感兴趣的分布。它是根据新的证据 y_N 更新先验信念

101 $p(r)$ 的结果。密度的形状是很有趣的——它告诉我们结合了已经知道的知识（先验）和观察到的知识（似然）之后，我们知道多少 r 的信息。图 3-5 给出了三个假设的示例（这些都是纯粹说明性的，并不对应图 3-3 和图 3-4 中特定的似然值和先验示例）。(a) 是均匀的——将似然值和先验结合使 r 的所有值是等可能的。(b) 表明 r 开始较小然后变大，这可能是开始的先验知识是均匀的，然后观察到更多的反面而不是正面。(c) 表明硬币倾向于正面朝上。因为这是密度，所以后验不是告诉我们哪个值是可能的，而是提供当有了这些观察数据后我们对 r 不确定程度的信息。

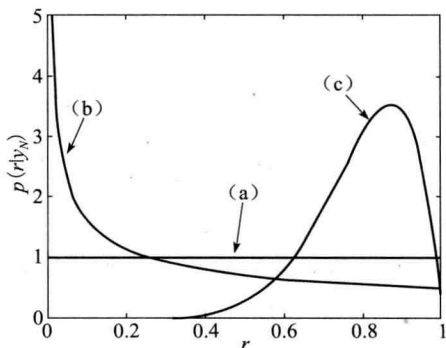


图 3-5 三个可能后验分布 $p(r|y_N)$ 的示例

$$E_{p(r|y_N)} \{P(Y_{10} \leq 6)\} = \int_{r=0}^{r=1} P(Y_{10} \leq 6 | r) p(r|y_N) dr$$

我们将会获得概率的期望值。考虑我们已经观察到的数据和我们的先验信念及保持的不确定性。这将有助于我们决定是否参与游戏。我们后面讨论这个问题，现在我们看看在硬币示例中获得的后验密度的类型。

3.2 精确的后验

当似然值是二项分布时， β 分布是先验的通常选择。这是因为可以用一些代数来精确地计算后验密度。事实上， β 分布是二项似然值的共轭先验（见注解 3.1）。如果先验和似然是共轭的，后验将和先验具有相同的形式。具体地， $p(r|y_N)$ 将给出一个 β 分布，参数 δ 和 γ 的值从先验和 y_N 中计算得到。 β 分布和二项分布不是唯一的共轭对，在本章奥运会数据中将会看到其他的先验和似然共轭对。

102 **注解 3.1（共轭先验）：**似然-先验对是共轭的，如果后验与先验具有相同的形式。这使我们能够分析计算后验密度，而不用关心计算贝叶斯规则的分母和边缘似然值。下表给出了一些常用的共轭对。

先验	似然
高斯	高斯
β	二项
γ	高斯
狄利克雷	多维正态

从数学的观点看，使用共轭先验使得事情变得更加容易。然后，正如我们在第 1 章讨论损失函数和第 2 章讨论的噪声分布那样，将我们的选择基于模型假设比基于数学方法更重要。在第 4 章中，我们将会看到一些用于不是共轭通常场景中的技术。

回到我们的例子，将 $p(y_N)$ 从式 (3-3) 中删除，得：

$$p(r|y_N) \propto P(y_N | r) p(r)$$

用二项分布和 beta 分布代替右边的项，得：

$$p(r|y_N) \propto \left[\binom{N}{y_N} r^{y_N} (1-r)^{N-y_N} \right] \times \left[\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1} (1-r)^{\beta-1} \right] \quad (3-5)$$

因为先验和似然是共轭的，所以我们知道 $p(r|y_N)$ 一定是 beta 密度。具有参数 δ 和 γ 的 beta 密度具有下面的通用形式：

$$p(r) = Kr^{\delta-1}(1-r)^{\gamma-1}$$

其中 K 是一个常量。如果我们可以将包括 r 在内的所有项移至式 (3-5) 的右侧使其看起来像 $r^{\delta-1}(1-r)^{\gamma-1}$ ，那么能够确定这一常量的正确性（它必须是 $\Gamma(\delta+\gamma)/\Gamma(\delta)\Gamma(\gamma)$ ），因为我们知道后验密度是 β 密度。另一方面，我们知道 β 密度的标准化常量，所以我们没有必要计算 $p(y_N)$ 。重新组织式 (3-5)，得：

$$\begin{aligned} p(r|y_N) &\propto \left[\binom{N}{y_N} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \right] \times [r^{y_N} r^{\alpha-1} (1-r)^{N-y_N} (1-r)^{\beta-1}] \\ &\propto r^{y_N+\alpha-1} (1-r)^{N-y_N+\beta-1} \\ &\propto r^{\delta-1} (1-r)^{\gamma-1} \end{aligned}$$

其中 $\delta = y_N + \alpha$ ， $\gamma = N - y_N + \beta$ 。

因此

$$p(r|y_N) = \frac{\Gamma(\alpha+\beta+N)}{\Gamma(\alpha+y_N)\Gamma(\beta+N-y_N)} r^{\alpha+y_N-1} (1-r)^{\beta+N-y_N-1} \quad (3-6)$$

103

（注意当增加 r 和 β 后， y_N 项就取消了）。这就是基于先验 $p(r)$ 和数据 y_N 的 r 的后验密度。注意后验参数是怎么计算的，通过给第一个参数 α 增加正面的数量（ y_n ）、给第 2 个参数 β 增加背面的数量（ $N - y_N$ ）。这允许我们获得一些关于先验参数 α 和 β 的直觉——它们可以看做是在 $\alpha + \beta$ 投币中正面朝上和背面朝上的次数。例如，考虑前一节讨论的第二个场景。对于公平的硬币场景， $\alpha = \beta = 50$ 。这等价于投了 100 次硬币，正面朝上和背面朝上的次数都是 50。对于有偏的场景， $\alpha = 5$ 、 $\beta = 1$ ，对应着 6 次投币有 5 次正面朝上。图 3-4 有助于我们解释两个密度表明的不同变化层次：公平投币密度相对于有偏投币具有较小的变化，因为它是较多假设投币的结果。投币次数越多，我们对 r 知道得就越多。

类推是不完美的。例如， α 和 β 不一定是整数，可以是小于 1 的数（0.3 正面朝上没有任何意思）。当 $\alpha = \beta = 1$ 时，类推也是不对的。观察到一次正面朝上和一次背面朝上表明 $r = 0$ 和 $r = 1$ 是不可能的。然而，图 3-4 的密度 1 表明 r 的所有取值是等可能的。尽管有这些不足，但类推是需要记住的，因为我们是通过分析前进的（见练习 EX 3.1、EX 3.2、EX 3.3 和 EX 3.4）。

3.3 三个场景

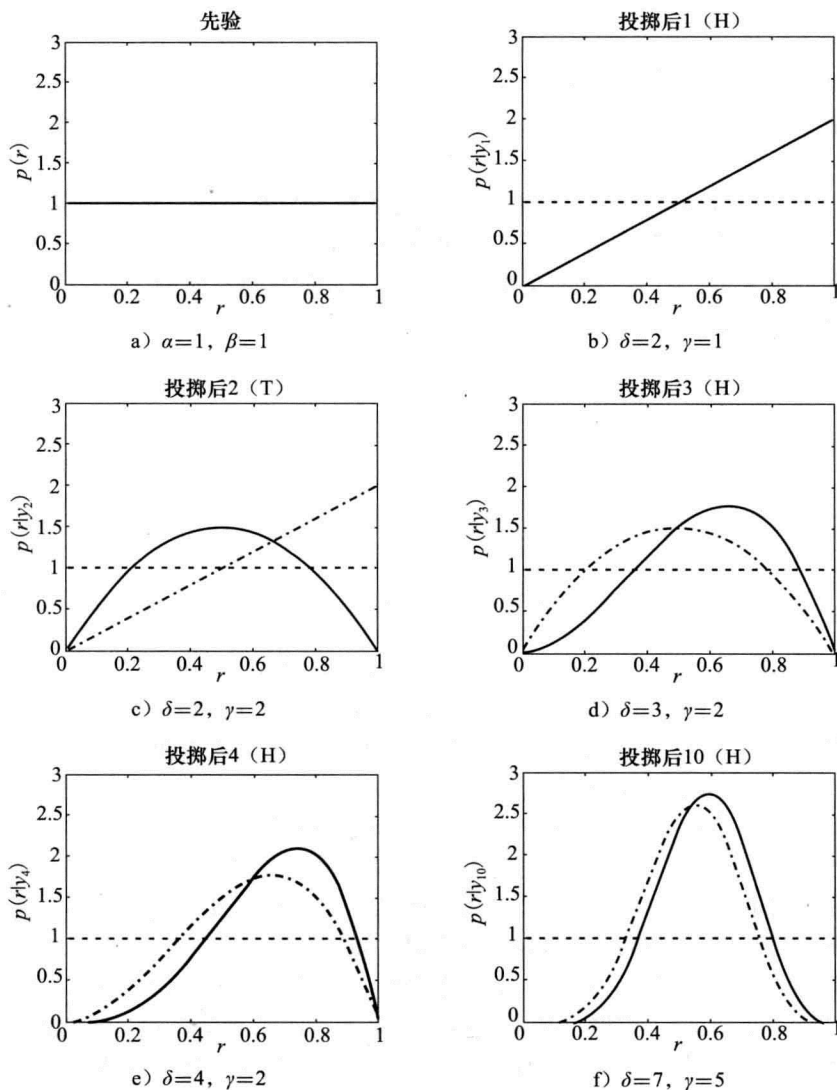
我们现在研究图 3-4 给出的三个不同场景的后验分布 $p(r|y_N)$ ，分别为没有先验知识、公平的投币和有偏的投币。

3.3.1 没有先验知识

在这个场景中（MATLAB 脚本：coin_scenario1.m），我们假设我们不知道硬币抛掷或货摊主的信息。我们的先验参数为 $\alpha = \beta = 1$ ，如图 3-6a 所示。

为了比较不同的场景，我们使用在先验下 r 的期望值和方差。具有参数 α 和 β 的 β 分布的随机变量（密度函数表示为 $B(\alpha, \beta)$ ）的期望值为（见练习 EX 3.5）：

$$\begin{aligned} p(r) &= B(\alpha, \beta) \\ E_{p(r)}\{R\} &= \frac{\alpha}{\alpha + \beta} \end{aligned}$$

图 3-6 随投币次数的增加 $p(r|y_N)$ 的变化过程

对场景 1:

$$E_{p(r)}\{R\} = \frac{\alpha}{\alpha + \beta} = \frac{1}{2}$$

β 分布随机变量的方差为 (见练习 EX 3.6):

$$\text{var}\{R\} = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} \quad (3-7)$$

当 $\alpha = \beta = 1$ 时, 有

$$\text{var}\{R\} = \frac{1}{12}$$

注意, 在我们的后验公式中 (式 (3-6)), 我们没有限制投币次数为 10——我们可以包含任何次数的投币。为了说明后验的演变, 我们看看它如何随投币变化。

一个新顾客交出了 1 镑, 货摊主开始投币。第一次投币结果是正面朝上。经过一次投币后, 后验分布是一个 β 分布, 参数 $\delta = \alpha + y_N$ 、 $\gamma = \beta + N - y_N$:

$$p(r|y_N) = \mathcal{B}(\delta, \gamma)$$

在这个场景中, $\alpha=\beta=1$, 且有 $N=1$ 投硬币, 看见 $y_N=1$ 次正面,

$$\delta = 1 + 1 = 2$$

$$\gamma = 1 + 1 - 1 = 1$$

图 3-6b 的实线为后验分布 (虚线为先验分布)。这个观察有相当大的作用——后验非常不同于先验。在先验中, r 的所有值是等可能的。现在发生了变化——零密度 $r=0$ 时, 高值比低值的可能性大。这与证据是一致的——观察到一个正面朝上使得 r 取大值的可能性高于 r 取小值。这个密度仍然是比较宽的, 因为我们只观察了一次投币。后验下 r 的期望值为:

$$E_{p(r|y_N)}\{R\} = \frac{2}{3}$$

同时我们观察到, 一个正面朝上使得 r 的期望值从 $1/2$ 变为 $2/3$ 。后验的方差为 (见式 (3-7)) 为:

$$\text{var}\{R\} = \frac{1}{18}$$

它小于先验方差 ($1/2$)。因此, 方差的减少告诉我们对 r 值的不确定性正在减少 (我们已经学到了一些), 期望值的增加告诉我们正面朝上比背面朝上的可能性大。

货摊主投第二枚硬币, 结果为背面朝上。我们已经看到一个正面朝上和一个背面朝上, 因此 $N=2$ 、 $y_N=1$, 结果为:

$$\delta = 1 + 1 = 2$$

$$\gamma = 1 + 2 - 1 = 2$$

图 3-6c 中的实线给出了后验分布。轻点画线是投币一次后的后验, 虚线是先验。密度已经再一次变化反映了新的证据。因为我们已经观察到了一个背面朝上, 所以 $r=1$ 的密度应该为 0 ($r=1$ 表示硬币总是正面朝上)。密度是曲线而不是直线 (我们已经提到, β 密度函数是非常灵活的), 观察一个背面朝上使小值的可能性提高了。期望值和方差现在为:

$$E_{p(r|y_N)}\{R\} = \frac{1}{2}, \text{var}\{R\} = \frac{1}{20}$$

期望值已经降到了 $1/2$ 。假设先验的期望值也是 $1/2$, 你可以得出结论: 我们没有学到任何知识。然而, 方差也降低了 (从 $1/18$ 到 $1/20$), 因此 r 的不确定性降低了, 已经学到了一些知识。实际上, 学到 r 比先验中假设的更接近 $1/2$ 。

第 3 次投币结果为正面朝上, 这时我们有 $N=3$ 、 $y_N=2$ 、 $N-y_N=1$ 次背面朝上。我们更新后验参数为:

$$\delta = \alpha + y_N = 1 + 2 = 3$$

$$\gamma = \beta + N - y_N = 1 + 3 - 1 = 2$$

图 3-6d 描绘了后验。后验是实黑线, 以前的后验是实轻线, 虚线为先验。我们注意到, 第 2 次正面朝上的观察效果使密度向右偏了, 表明正面朝上的可能性高于背面朝上。显然这与观察是一致的——我们看过的正面比背面多。我们只看了 3 个硬币, 然而这仍然存在很高的不确定性, 密度表明 r 可以取 $0 \sim 1$ 内的很多值。新的期望和方差为:

$$E_{p(r|y_N)}\{R\} = \frac{3}{5}, \text{var}\{R\} = \frac{1}{25}$$

方差又一次减小, 表明随着观察数据的增多不确定性减小。

第 4 次投币也是正面 ($y_N=3$, $N=4$), 则 $\delta=\alpha+y_N=1+3=4$, $\gamma=\beta+N-y_N=1+4-3=2$ 。图 3-6e 给出了以前和现在的后验和先验。密度又一次向右偏——我们现在看到 3 次正面、一次背面, 因此这看起来好像 r 大于 $1/2$ 。注意 $N=3$ 的后验和 $N=4$ 的后验 r 取非常小的值时的差别, 这次正面使我们想到 r 不会等于或小于 0.1。期望值和方差为:

$$E_{p(r|y_N)}\{R\} = \frac{2}{3}, \text{var}\{R\} = \frac{2}{63} = 0.0317$$

其中期望值增大了, 方差再一次减小。其余 6 次投币完成后, 整个投币序列为:

H, T, H, H, H, H, T, T, T, H

6 个正面、4 个背面。当 $N=10$ 、 $y_N=6$ 时, 后验分布参数为: $\delta=\alpha+y_N=1+6=7$ 、 $\gamma=\beta+N-y_N=1+10-6=5$ 。该分布和 $N=9$ 时的后验如图 3-6f 所示。期望值和方差为:

$$E_{p(r|y_N)}\{R\} = \frac{7}{12} = 0.5833, \text{var}\{R\} = 0.0187 \quad (3-8)$$

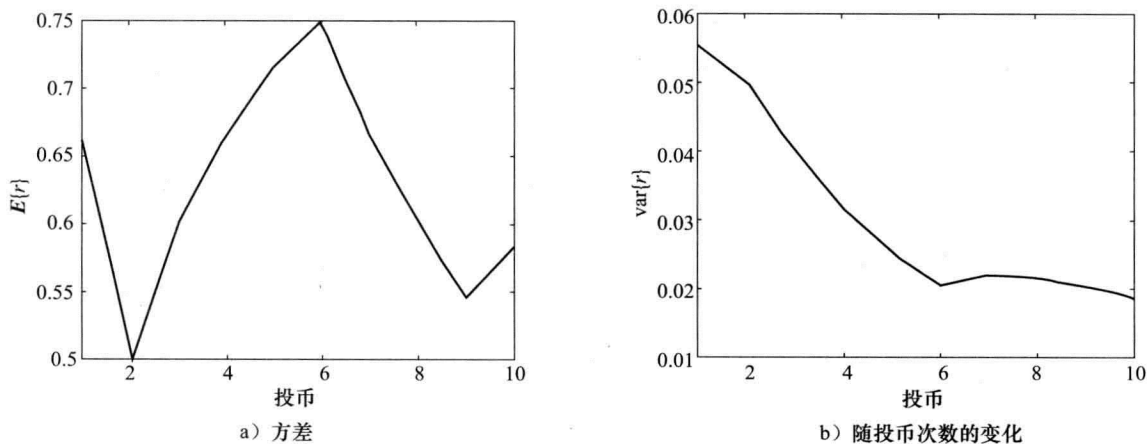


图 3-7 参数 r 的期望

经过 10 次观察, 期望值从 0.5 增加到了 0.5833, 方差从 $1/12=0.0833$ 减小了 0.0187。然而, 这并不是整个故事。观察图 3-6f, 我们看到并非常确定 $r>0.2$ 且 $r<0.9$ 。这时 r 的不确定性还很高的, 因为我们只观察了 10 次投币。

图 3-7 总结了期望值和方差随着 10 次观察的变化。期望值变化较小, 而方差随着信息的增多稳步减小。在第 7 次投币时, 方差增加。最初的 7 次投币为:

H, T, H, H, H, H, T

当第 6 次投币结束时, 有 5 次正面、一次背面, 这表明正面的可能性大于背面。第 7 次出现的背面是不希望的。图 3-8 给出了第 7 次前和后的后验。反面事件导致密度函数增加 r 取小值的似然, 因此增加了不确定性。

后验密度概括了所有 r 的信息。简单地说, 我们使用它来计算游戏获胜的期望概率。在做这之前, 我们将重新回到使用点估计的想法, 从密度中提取 r 的单个值 \hat{r} 。我们将比较获胜的期望概率与从 r 的单个值计算出的获胜的概率。一个合理的选择是使用 $E_{p(r|y_N)}\{R\}$ 。有了这个值, 我们就能够计算获胜的概率—— $P(Y_{\text{new}} \leq 6 | \hat{r})$ 。这个值用于决定是否玩游戏。注意区别观察的投币和未来的投币, 我们使用随机变量 Y_{new} 来描述 10 次未来的投币。

10 次投币后, 后验密度是 β 概率,

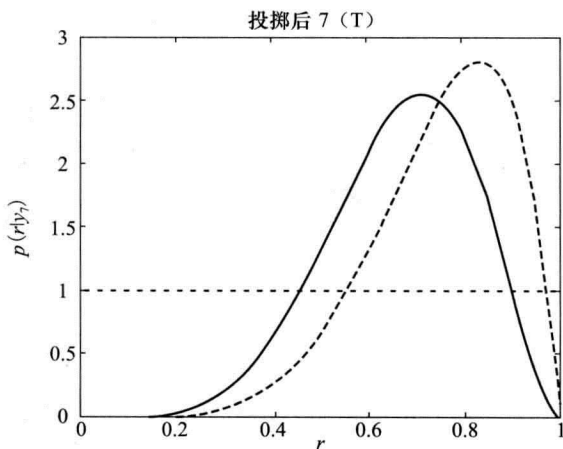


图 3-8 6 次投币 (虚) 和 7 次投币 (实) 的后验

参数 $\delta=7$ 、 $\gamma=5$ 。因此 \hat{r} 为:

$$\hat{r} = \frac{\delta}{\delta + \gamma} = \frac{7}{12}$$

游戏获胜的概率为:

$$\begin{aligned} P(Y_{\text{new}} \leq 6 | \hat{r}) &= 1 - \sum_{y_{\text{new}}=7}^{10} P(Y_{\text{new}} = y_{\text{new}} | \hat{r}) \\ &= 1 - 0.3414 \\ &= 0.6586 \end{aligned}$$

表明我们获胜的可能高于失败的可能。

使用所有的后验信息需要计算,

$$\mathbf{E}_{p(r|y_N)} \{P(Y_{\text{new}} \leq 6 | r)\}$$

重新组织和操作期望, 得:

$$\begin{aligned} \mathbf{E}_{p(r|y_N)} \{P(Y_{\text{new}} \leq 6 | r)\} &= \mathbf{E}_{p(r|y_N)} (1 - P(Y_{\text{new}} \geq 7 | r)) \\ &= 1 - \mathbf{E}_{p(r|y_N)} \{P(Y_{\text{new}} \geq 7 | r)\} \\ &= 1 - \mathbf{E}_{p(r|y_N)} \left\{ \sum_{y_{\text{new}}=7}^{y_{\text{new}}=10} P(Y_{\text{new}} = y_{\text{new}} | r) \right\} \\ &= 1 - \sum_{y_{\text{new}}=7}^{y_{\text{new}}=10} \mathbf{E}_{p(r|y_N)} \{P(Y_{\text{new}} = y_{\text{new}} | r)\} \end{aligned} \quad (3-9)$$

为了评估它, 我们必须能够计算 $\mathbf{E}_{p(r|y_N)} \{P(Y_{\text{new}} = y_{\text{new}} | r)\}$ 。根据期望的定义, 得:

$$\begin{aligned} \mathbf{E}_{p(r|y_N)} \{P(Y_{\text{new}} = y_{\text{new}} | r)\} &= \int_{r=0}^{r=1} P(Y_{\text{new}} = y_{\text{new}} | r) p(r | y_N) dr \\ &= \int_{r=0}^{r=1} \left[\binom{N_{\text{new}}}{y_{\text{new}}} r^{y_{\text{new}}} (1-r)^{N_{\text{new}}-y_{\text{new}}} \right] \left[\frac{\Gamma(\delta+\gamma)}{\Gamma(\delta)\Gamma(\gamma)} r^{\delta-1} (1-r)^{\gamma-1} \right] dr \\ &= \binom{N_{\text{new}}}{y_{\text{new}}} \frac{\Gamma(\delta+\gamma)}{\Gamma(\delta)\Gamma(\gamma)} \int_{r=0}^{r=1} r^{y_{\text{new}}+\delta-1} (1-r)^{N_{\text{new}}-y_{\text{new}}+\gamma-1} dr \end{aligned} \quad (3-10)$$

这个式子看起来有点让人畏惧。然而, 仔细观察式子中的参数是一个没有标准化的 β 密度, 参数为 $\delta+y_{\text{new}}$ 和 $\gamma+N_{\text{new}}-y_{\text{new}}$ 。一般来说, 参数为 α 和 β 的 β 密度必须满足下式:

$$\int_{r=0}^{r=1} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1} (1-r)^{\beta-1} dr = 1$$

因此:

$$\int_{r=0}^{r=1} r^{\alpha-1} (1-r)^{\beta-1} dr = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

我们的期望值变为:

$$\mathbf{E}_{p(r|y_N)} \{P(Y_{\text{new}} = y_{\text{new}} | r)\} = \binom{N_{\text{new}}}{y_{\text{new}}} \frac{\Gamma(\delta+\gamma)}{\Gamma(\delta)\Gamma(\gamma)} \frac{\Gamma(\delta+y_{\text{new}})\Gamma(\gamma+N_{\text{new}}-y_{\text{new}})}{\Gamma(\delta+\gamma+N_{\text{new}})}$$

对一个特定的后验 (也就是, γ 和 δ 的值) 和 N_{new} 及 y_{new} 的值, 可以很容易计算上式。

10 次投币后, 我们有 $\delta=7$ 、 $\gamma=5$ 。将这些值代入, 我们能够计算获胜的期望概率:

$$\begin{aligned} \mathbf{E}_{p(r|y_N)} \{P(Y_{\text{new}} \leq 6 | r)\} &= 1 - \sum_{y_{\text{new}}=7}^{y_{\text{new}}=10} \mathbf{E}_{p(r|y_N)} \{P(Y_{\text{new}} = y_{\text{new}} | r)\} \\ &= 1 - 0.3945 \\ &= 0.6055 \end{aligned}$$

比较这个值与点估计的值, 我们能够看到这两个预测都是获胜的机会大。这也与证据一

致——一个人获得 6 次正面、4 次背面，因此赢得 2 镑。点估计给出了较高的概率——忽略后验的不确定性使得它有可能认为我们会赢。

另外一个顾客玩这个游戏，投币的序列为：

H, H, T, T, H, H, H, H, H, H

8 次正面、2 次背面——货摊主赢了。结合 20 次投币，我们有 $N=20$ 、 $y_N=6+8=14$ 次正面， $N-y_N=20-14=6$ 次背面。则得出 $\delta=15$ 、 $\gamma=7$ 。图 3-9 给出了后验密度，其中细线给出了 10 次投币后的后验，虚线为先验。期望值和方差为：

110

$$E_{p(r|y_N)}\{R\} = 0.6818, \text{var}\{R\} = 0.0094$$

期望值增加了，方差减小了（参见式 (3-8)）。这都在我们的预料之中——8 次正面和 2 次背面应该能增加 r 的期望值，增加的数据应该能降低方差。

依据新的证据，我们现在能重新计算 $E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6 | r)\}$ 。代入适应的值，得：

$$E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6 | r)\} = 0.4045$$

新的证据使密度向右偏离，使 r 值增大（硬币正面朝上）的可能性提高，降低了获胜的可能性。

为了完整性，我们还计算了 $P(Y_{\text{new}} \leq 6 | \hat{r}) = 0.3994$ 。

它对应的期望收益为：

$$2 \times 0.4045 - 1 = -0.1910$$

等价于玩一次游戏损失 20 便士。

在这个例子中，我们已经接触到贝叶斯机器学习的所有重要部分——选择先验、选择似然、计算后验、使用期望进行预测。我们将在另外两个先验场景下重复这个过程。

3.3.2 公平的投币

对于公平的投币场景（MATLAB 脚本：coin_scenario2.m），我们假设 $\alpha=\beta=50$ ，这类似于我们投了 100 次硬币，恰好有一半的硬币正面朝上。需要注意的第一件事是，100 次投币比我们刚才观察的 20 次投币具有更多的数据。我们应该期望我们的数据和前一个场景具有相同的效果吗？

111

图 3-10a 给出了先验密度，图 3-10b、c、d、e 和 f 分别给出了投币次数为 1、5、10、15 和 20 次的后验。对于这个场景，在每个阶段我们没有给出以前的后验——它与当前的后验太近了。然而，在大多数情况下，后验的变化太小以至于两条线几乎重合。实际上，10 次投币以后，后验已经和先验不同了。回想我们对 β 先验的类推，先验知识包括 100 次投掷硬币的证据，因此增加 10 次投币后会有区别是不令人惊奇的。

$E_{p(r|y_N)}\{R\}$ 和 $\text{var}\{R\}$ 随着 20 次投币的演变过程见图 3-11。在数据出现之后，图 3-11 与图 3-6 相比只有非常小的变化。这个小变化表明了非常强的先验密度。这个先验会控制数据直到我们观察了许多次投币——也就是，在式 (3-3) 中 $p(r)$ 控制 $p(y_N | r)$ 。我们创造了一个模型，这个模型目前停滞不前，并且需要更多的说服力来信任或者否定它。

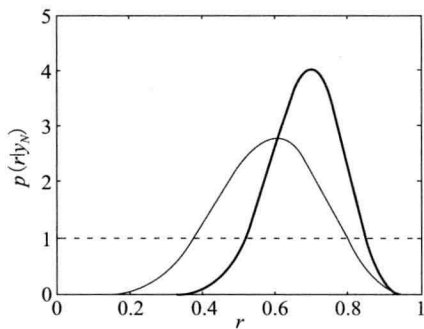
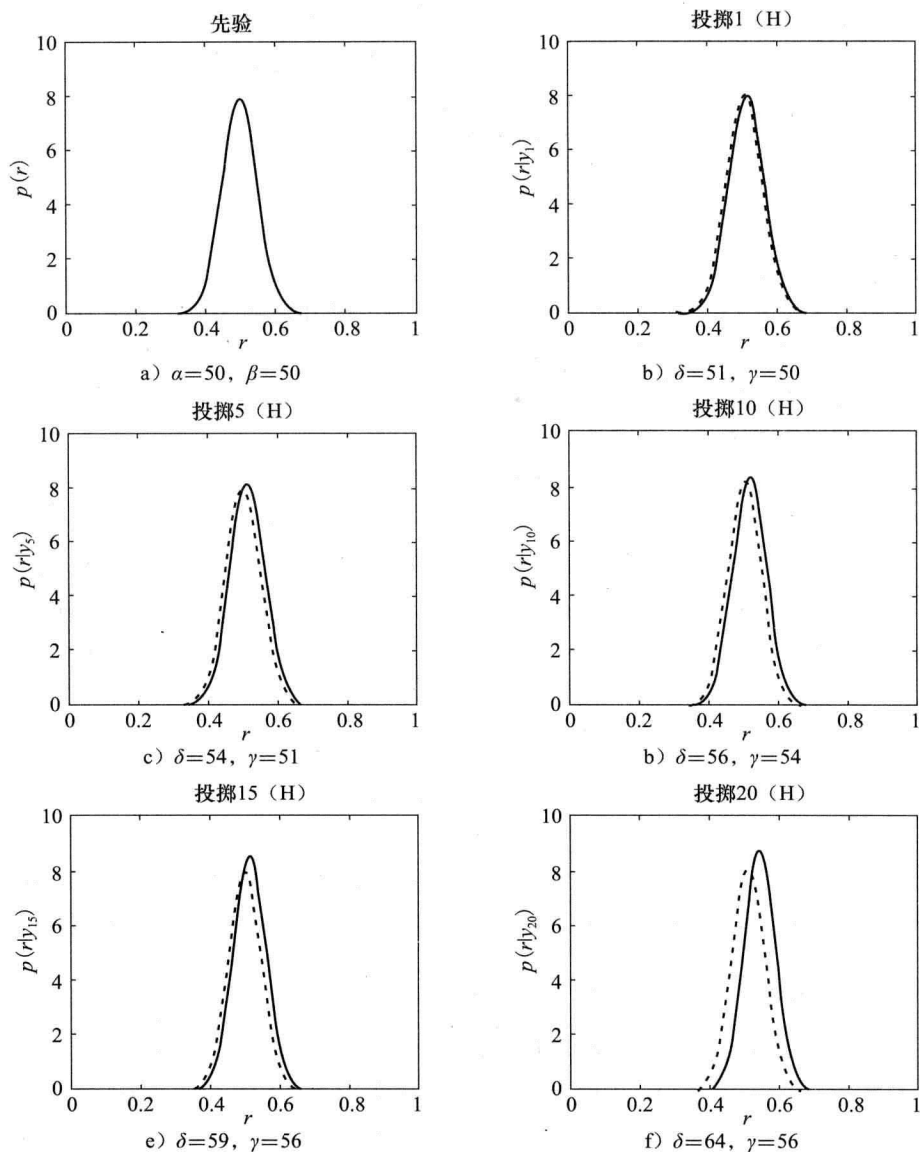
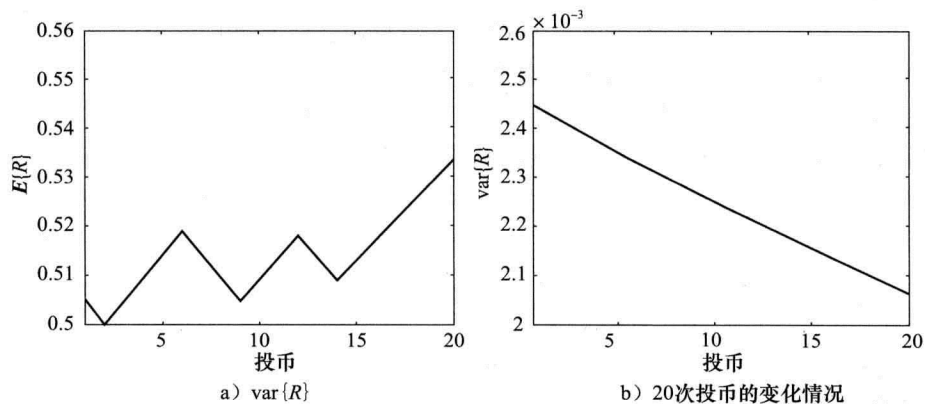


图 3-9 10 次投币（虚曲线）和 20 次投币（实曲线）的后验分布，虚线对应于先验密度


 图 3-10 公平的投币场景下，后验 $p(r|y_N)$ 随投币次数的变化情况。虚线为先验密度

 图 3-11 公平的投币场景下， $E_{p(r|y_N)}\{R\}$ 的估计值

像前一节一样,我们能计算出 $E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6|r)\}$ 。经过 20 次投币后,我们有 $\delta = \alpha + y_N = 50 + 14 = 64$ 、 $\gamma = \beta + N - y_N = 50 + 20 - 14 = 56$ 。期望值计算如下:

$$E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6|r)\} = 0.7579 \quad (3-11)$$

像前面一样,我们也可以看到该值与使用点估计 \hat{r} 得到的值 $P(Y_{\text{new}} \leq 6|\hat{r})$ 不同(在这种情况下, $\hat{r} = 64/(64+56) = 0.5333$):

$$P(Y_{\text{new}} \leq 6|\hat{r}) = 0.7680$$

这两个量预测我们将获胜。依据我们看到的后验信息,这个结论是不令人惊讶的。这个数据已经有些克服了先验假设:投币是公平的,并且我们已经知道如果投币是公平的,我们将会赢(公平的投币会导致我们赢,平均情况下,每次游戏赢 66 便士,见 3.1 节)。

作为一个方面,考虑这个场景和前一个场景中我们的近似 $P(Y_{\text{new}} \leq 6|\hat{r})$ 与适当期望值的精确度。在前一个场景中,这两个值之间的差为:

$$|E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6|r)\} - P(Y_{\text{new}} \leq 6|\hat{r})| = 0.0531 \quad [113]$$

在本场景中,这两个值是比较接近的:

$$|E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6|r)\} - P(Y_{\text{new}} \leq 6|\hat{r})| = 0.0101$$

这是出现下面情况很好的原因——随着后验方差的减小(场景 2 的方差小于场景 1),概率密度变得越来越集中在一个点的附近。想象方差降低到这样一个程度, r 只取一个值时,出现 $p(r|y_N)$ 的概率为 1,取其他值时概率为 0。期望值计算如下:

$$E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6|r)\} = \int_{r=0}^{r=1} P(Y_{\text{new}} \leq 6|r) p(r|y_N) dr$$

如果除了在特定值 \hat{r} 外的其他位置 $p(r|y_N)$ 都为 0,这就变为:

$$E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6|r)\} = P(Y_{\text{new}} \leq 6|\hat{r})$$

换句话说,随着方差的减小, $P(Y_{\text{new}} \leq 6|\hat{r})$ 变得越来越近似于真实期望值。这不是这个示例的特殊情况——随着数据量的增加(参数的不确定性降低),点估计就变得更可靠了。

3.3.3 有偏的投币

在最后一个场景中,我们假设硬币(和货摊主)偏向于正面朝上而不是背面朝上(MATLAB 脚本: coin_scenario3.m)。将它编码为一个 β 先验,参数 $\alpha=5$ 、 $\beta=1$ 。期望值为:

$$E_{p(r)}\{r\} = 5/6$$

投币 6 次,有 5 次正面。正如场景 2 一样,图 3-12a 给出了先验密度,图 3-12b、c、d、e 和 f 分别给出投币 1、5、10 和 20 次的后验。给定我们已经看到的,这里没有什么不平常的。后验相当快地远离了先验(先验有效地影响了 $\alpha+\beta=6$ 个数据点)。图 3-13 给出了期望值和方差的演变。方差曲线有多个起伏不平的点,这些点对应着投币结果为背面。这是因为强烈的先验偏向于高的 r 值。在这种假设下我们不期望看到多的背面,因此当出现背面时,模型变得不确定了。而且,我们计算感兴趣的真实量, $E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6|r)\}$ 。最后的后验参数为 $\delta = \alpha + y_N = 5 + 14 = 19$ 、 $\gamma = 1 + N - y_N = 1 + 20 - 14 = 7$ 。代入,得:

$$E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6|r)\} = 0.2915$$

注意 $\hat{r} = 19/(19+7) = 0.7308$,近似值为:

$$P(Y_{\text{new}} \leq 6|\hat{r}) = 0.2707$$

$$[114] \quad \text{这两个值表明平均来说我们会输,即损失钱。}$$

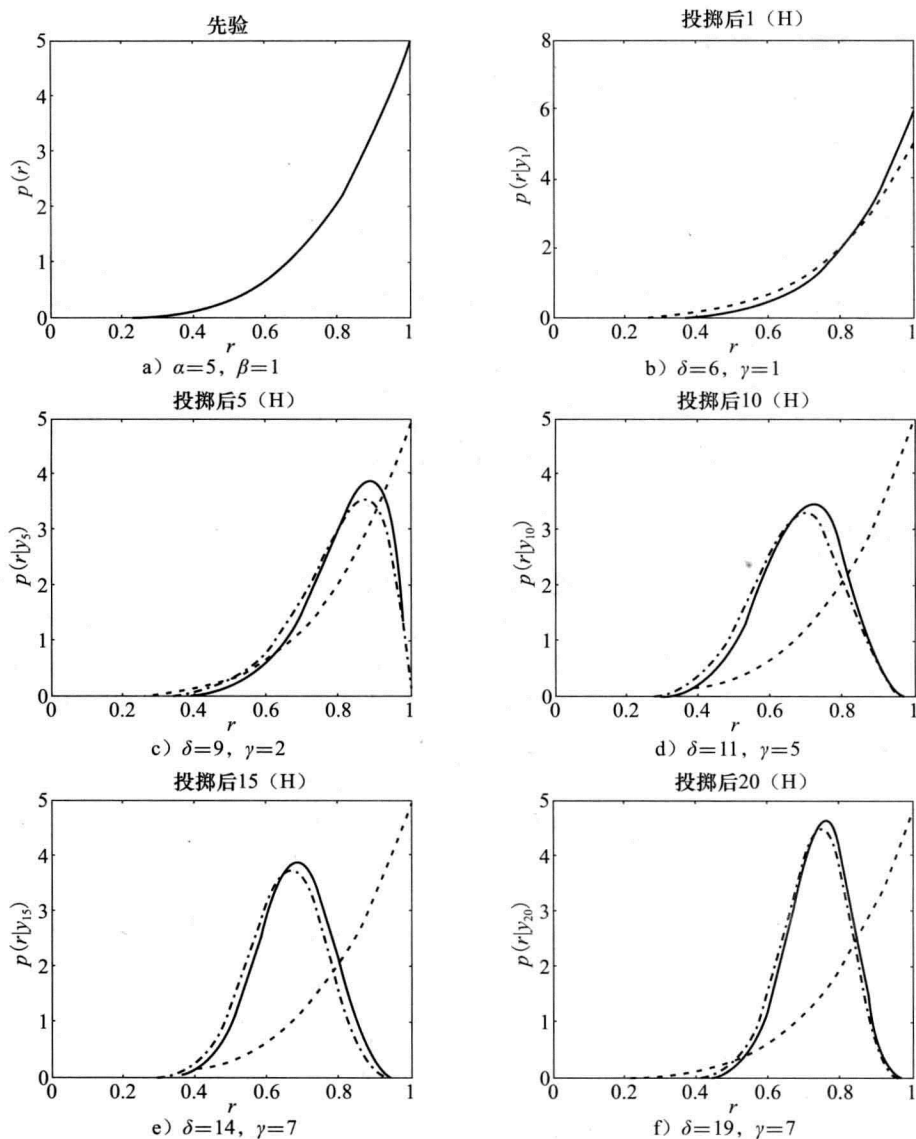


图 3-12 有偏的投币场景下后验 $p(r|y_N)$ 随更多投币次数的变化。虚线是先验密度，最后 4 张图的点画线是上一次的后验（也就是 4 次、9 次、14 次和 19 次投币后的后验）

115

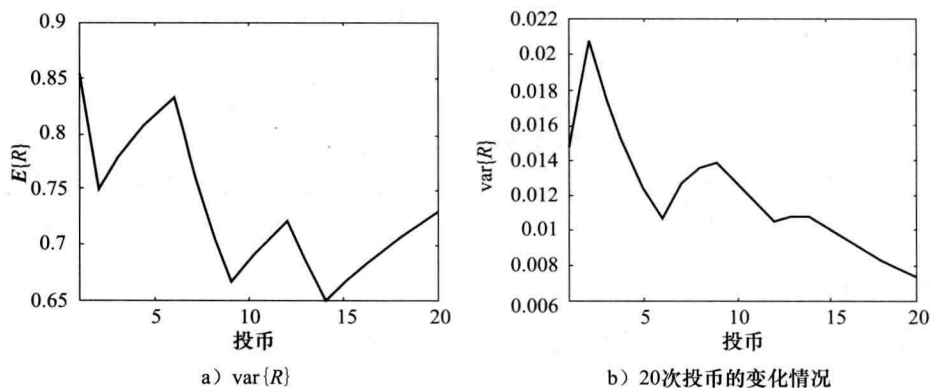


图 3-13 有偏的投币场景下的 $E_{p(r|y_N)}\{R\}$

3.3.4 三个场景——总结

我们的三个场景已经给出了获胜期望概率的不同值：

1) 没有先验知识： $E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6 | r)\} = 0.4045$ 。

2) 公平的投币： $E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6 | r)\} = 0.7579$ 。

3) 有偏的投币： $E_{p(r|y_N)}\{P(Y_{\text{new}} \leq 6 | r)\} = 0.2915$ 。

我们应该选择哪个呢？我们依据先验信念进行选择看似是合理的。如果货摊主没有看起来要失业，那么场景3可能是合理的。我们可能决定我们确实不知道关于货摊主和硬币的任何事情，然后看看场景1。我们可能认为一个正直的货摊主从来不会骗人，因此进入场景2。为它们中的任何一个找出理由都是可能的。可以看到，贝叶斯技术允许我们从理论上将观测到的数据（20次掷硬币）和某些先验知识（上述某一场景）结合起来。后验概率密度明确地模拟了每个阶段 r 的不确定性，并且可以用来做出预测。（见练习EX 3.7和EX 3.8）。

3.3.5 增加更多的数据

在我们继续讨论之前，检查增加更多数据后的效果是值得的。我们看到在每个场景里增加更多的数据导致后验与先验的分歧——通常通过方差的减小。实际上，如果我们继续增加数据，我们就会发现三种场景的后验开始看起来非常相似。图3-14给出了投币次数为100和1000时三种场景的后验。将三种场景投少量硬币后的后验（见图3-6f、图3-10d、图3-12d）进行比较，我们发现后验变得越来越相似。尤其是投币次数为1000时，场景1和场景3变得没法区分了。场景1和场景3与场景2的区别在于场景2先验的方差很小——先验对应于非常强的信念，这需要非常多的反数据来移除这种影响。

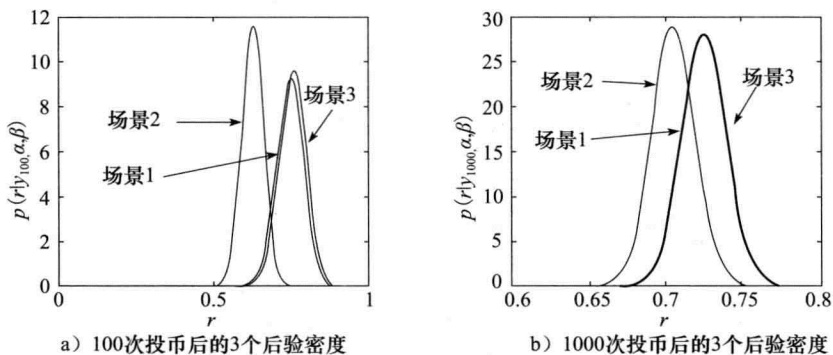


图 3-14 100 次投币后 a) 和 1000 次投币后 b) 三个场景的后验密度

随着数据的增加，先验影响的减少是很容易解释的，如果我们看看用于计算后验的表达式。忽略标准化边缘似然项，后验正比于似然与先验的乘积。当我们增加更多的数据时，先验不发生变化，但似然变成了个体似然的乘积（如果通常的独立假设成立）。这个增加会逐渐压倒来自先验的单个分布。这是非常直观的——当我们观察越来越多的数据时，在观察之前的信念会变得越来越不重要了。

3.4 边缘似然估计

幸运的是，从上述三个方案中选取最佳方案的标准并非只取决于主观臆断。在上述章节中，从式(3-3)可以得知，分母 $p(y_N)$ 与 r 间的关系为：

$$\begin{aligned}
 p(y_N) &= \int_{r=0}^{r=1} p(r, y_N) dr \\
 &= \int_{r=0}^{r=1} p(y_N | r) p(r) dr
 \end{aligned} \quad (3-12)$$

为了确定 $p(r)$ 的值, 需要进一步约束条件。通常, $p(r)$ 应该写为 $p(r|\alpha, \beta)$, 即应看做是给定参数对 α 和 β 的条件概率密度。此时, 式 (3-12) 转换为如下形式:

$$p(y_N | \alpha, \beta) = \int_{r=0}^{r=1} p(y_N | r) p(r | \alpha, \beta) dr \quad (3-13)$$

边缘似然估计 (之所以如此称呼是因为 r 已被边缘化了) $p(y_N | \alpha, \beta)$ 是一个至关重要的值。这说明了通过预先给定参数对 α 和 β , 可以确定数据 y_N 的可能值 (或者出现数据 y_N 的概率)。 $p(y_N | \alpha, \beta)$ 的值越高, 越符合先验分布。因此, 对于数据集, 可以使用 $p(y_N | \alpha, \beta)$ 选择最佳场景, 即选择 $p(y_N | \alpha, \beta)$ 值最高的那个方案。

为了获得这个值, 需要按照如下的积分形式进行计算:

$$\begin{aligned}
 p(y_N | \alpha, \beta) &= \int_{r=0}^{r=1} p(y_N | r) p(r | \alpha, \beta) dr \\
 &= \int_{r=0}^{r=1} \binom{N}{y_N} r^{y_N} (1-r)^{N-y_N} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1} (1-r)^{\beta-1} dr \\
 &= \binom{N}{y_N} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \int_{r=0}^{r=1} r^{\alpha+y_N-1} (1-r)^{\beta+N-y_N-1} dr
 \end{aligned}$$

这和式 (3-10) 的形式完全相同。可以看到, 此式内的积分部分是一个非规范化的 β 密度, 因此通过整合可以得到正态的 β 规范化常数的逆变换。因此

$$p(y_N | \alpha, \beta) = \binom{N}{y_N} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(\alpha+y_N)\Gamma(\beta+N-y_N)}{\Gamma(\alpha+\beta+N)} \quad (3-14)$$

在上述例子中, $N=20$, $y_N=14$ (两组 10 次抛币中共有 14 个正面)。取三组不同的 α 和 β 值, 代入式 (3-14) 中, 可得:

- 1) 无先验知识, $\alpha=\beta=1$, $p(y_N | \alpha, \beta)=0.0476$ 。
- 2) 公平的投币, $\alpha=\beta=50$, $p(y_N | \alpha, \beta)=0.0441$ 。
- 3) 有偏的投币, $\alpha=5$, $\beta=1$, $p(y_N | \alpha, \beta)=0.0576$ 。

有偏的投币的先验具有最高的边缘似然值, 公平的投币的先验具有最低值。通过前面章节的学习可以知道, 这种方案获胜的概率为 $E_{p(r|y_N, \alpha, \beta)} \{P(Y_{\text{new}} \leq 6 | r)\} = 0.2915$ (注意, 此时的后验条件是以先验参数 $p(r | y_N, \alpha, \beta)$ 为基础的)。

需要特别指出的是, 以这种方式选择的先验知识本质上是选择最切合数据的先验。先验不再与我们观测到数据之前的先验相对应。在某些实际应用中, 这可能是无法接受的。这给出了一个值来表示数据对于先验信念的支持程度。在上面的例子中, 数据说明有偏的投币的先验是最好的证据。

3.4.1 与边缘似然做模型比较

在前述章节中通过使用边缘似然来优化 α 和 β 的值来对先验进行估计是可行的。假设 α 和 β 的选取范围如下:

$$\begin{aligned}
 0 &\leq \alpha \leq 50 \\
 0 &\leq \beta \leq 30
 \end{aligned}$$

可以最大化 $p(y_N | \alpha, \beta)$ 的 α 和 β 值。

图 3-15 显示了当 α 和 β 在各自范围内取不同值时, 边缘似然估计对应的数值。其最优

值为 $\alpha=50$ 、 $\beta=22$ ，对应的边缘似然估计是 0.1694。以这种方式选择参数称为最大似然估计类型 II，这样标记主要是为了与第 2 章中所讲的标准最大似然估计（称为类型 I）相区分。

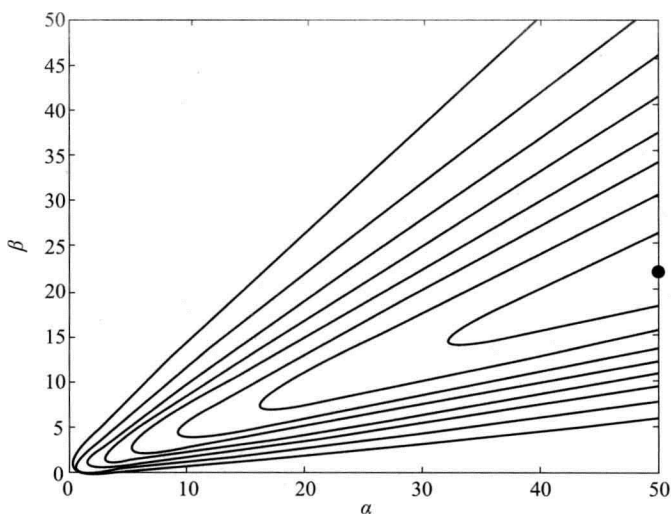


图 3-15 在抛硬币例子中，边缘似然估计的曲线图（即先验参数 α 、 β 的函数），最右上方的圆点表示最优值

3.5 超参数

从目前的研究来看，贝叶斯分析法允许将任意多个感兴趣的参数作为随机变量来进行分析处理（例如，抛硬币实验中正面朝上的次数 r ）。在此例子中， r 并非是唯一让我们感兴趣的参数。 α 和 β 也是我们感兴趣的参数——是否能够对它们进行类似的分析处理呢？在某些情况下，基于问题的知识（我们知道硬币是有偏的）可以直接对它们进行特值分析。通常，我们并不知道它们确切的取值，因此可以将它们作为随机变量来处理。为了达到此目的，需要定义一个基于所有随机变量的先验概率— $p(r, \alpha, \beta)$ 。其具体计算过程如下（详见 2.2.5 节）：

$$p(r, \alpha, \beta) = p(r|\alpha, \beta)p(\alpha, \beta)$$

另外，通常假定 α 和 β 是相对独立的（ $p(\alpha, \beta) = p(\alpha)p(\beta)$ ）非常有用。在此模型中，我们感兴趣的是包括所有参数的后验概率，即

$$p(r, \alpha, \beta|y_N)$$

根据贝叶斯规则，则有

$$\begin{aligned} p(r, \alpha, \beta|y_N) &= \frac{p(y_N|r, \alpha, \beta)p(r, \alpha, \beta)}{p(y_N)} \\ &= \frac{p(y_N|r)p(r, \alpha, \beta)}{p(y_N)} \\ &= \frac{p(y_N|r)p(r|\alpha, \beta)p(\alpha, \beta)}{p(y_N)} \end{aligned}$$

注意在第二步中，从似然 $p(y_N|r)$ 中移除了 α 和 β 。这是因为它们具有条件独立性（详见 2.7.1 节）。 y_N 的分布主要依靠 α 和 β ，但通过它们对 r 产生影响。如果给定 r 一个特值，这种独立性就会不成立。

$p(\alpha, \beta)$ 通常需要一些附加参数—— $p(\alpha, \beta|\kappa)$ ，如同 α 和 β 将对 r 产生影响， κ 也将对

α 和 β 产生影响。 κ 称为**超参数**，因为它是控制 r 的参数的参数。在计算边缘似然估计时，对所有的变量积分，只留下取决于超参数的数据：

$$p(y_N | \kappa) = \iiint p(y_N | r) p(r | \alpha, \beta) p(\alpha, \beta | \kappa) dr d\alpha d\beta$$

不幸的是，这将增加模型的复杂度对感兴趣的参数（后验概率 $p(r, \alpha, \beta | y_N, \kappa)$ 、任何预测期望和边缘似然估计 $p(y_N | \kappa)$ ）的计算难度将增大，需要一种近似分析方法来解决这个问题，此种近似分析方法将在第4章中进行详细讲解。

至此，可以想象无限层模型的情况。例如， κ 可以认为是基于其他随机变量密度的随机变量。模型的层数（确定一个或多个参数时的迭代关系）由建模使用的数据集（某些层可以详细说明的准确值）和所能承受的计算复杂度来决定。通常，层数越多，计算的复杂度越大，预测的也越准确。

3.6 图模型

当给模型增加额外的层时（如超参数等），模型将很快变得难以处理。通常使用图模型对其进行描述。**图模型**是一个网状图，节点对应于随机变量，边代表变量间的依赖关系，例如，在2.2.4节中，介绍了多种随机变量间的属性关系，这些属性关系是通过抛硬币模型中的两个变量来呈现的，即硬币的投掷结果 X 和猜测结果 Y 。此模型定义为条件分布 $P(Y = y | X = x)$ ，其关系如图3-16a所示。两个节点通过一个箭头线连接，表示 Y 是依赖条件 X 而定义的。注意，节点 Y 是有阴影的，这是因为只要听者足够仔细，这个变量就是可观测的。听者无需观测硬币的实际投掷结果，因此也无需观察 X 。假设投掷过程重复 N 次，则有 $2N$ 个随机变量 X_1, \dots, X_N 和 Y_1, \dots, Y_N 。全部画出这些变量是非常困难的。可以通过在**模板**中嵌入节点来解决这一问题。模板是一个矩形框所界定，矩形框中的数值表示此模板被重用的次数。此数值放在矩形框的右下角，如图3-16b所示。

120

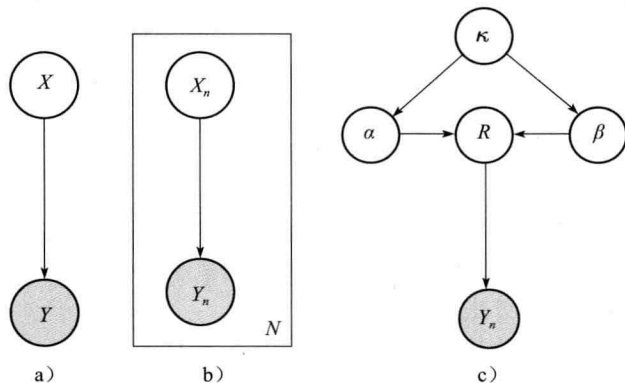


图 3-16 图模型例子。节点对应于随机变量，阴影节点为可观测的变量。箭头线描述了变量间的依赖关系，模板描述了多个实例。例如，在 b) 中，有 N 个随机变量 $Y_n (n=1, \dots, N)$ ，并且每个变量都依赖于随机变量 X_n 。c) 是抛硬币例子的图模型实例，是给出的硬币投掷模型的图模型表示。模型具有单一的观察变量表示 N 次投掷中证明的次数 Y_n 。它受变量 R 的影响，而 R 取决于随机变量 α 和 β 。最终 α 和 β 取决于超参数 κ 。

图 3-16c 显示了抛硬币模型中各个变量间的关系。在 N 次重复的投掷过程中，有一个单独的（可观测的）随机变量代表了出现正面的次数 y_N 。这是条件随机变量 R ，它依赖于随机变量 α 和 β ，而 α 和 β 则最终依赖于超参数 κ 。

建议读到本章末尾，以获得更多的关于图模型的信息。

3.6.1 小结

在前面的章节中,我们已经引入了许多新的概念,其中最重要的是应该将所有感兴趣的参数都看做随机变量来进行处理。为了做到此点,必须定义一个基于所有可能变量的先验分布,并使用贝叶斯规则(式(3-3))对其进行分析,以便得到在证据中并入观测数据后概率密度如何改变。可以对结果的后验概率进行检测并用于计算感兴趣的期望。另外,我们还讨论了如何使用边缘似然估计(贝叶斯规则中的正态常数)来计算不同的模型(例如,在抛硬币实验中选择最优先验分布),并讨论了此方法的缺陷和使用瓶颈。最后,讨论了如何通过将参数视为随机变量并定义其先验的方式扩展贝叶斯规则。像这样增加层次性将使得计算变得棘手,从而不得不求助于基于采样和近似的技术,这些方法将在第4章中进行详细的介绍。

3.7 奥运会 100 米数据的贝叶斯处理实例

现在返回到奥运会 100 米数据。在之前的章节中,通过将最小平方损失和明确的噪声模型相结合,拟合了一个线性(相对于参数而言)模型,并且使用最大似然估计找到了最优参数。在本节中,将使用贝叶斯处理方法,给出在伦敦举行的 2012 届奥运会的预测数据。这将包括多个步骤。首先,需要定义先验概率和似然估计(与在抛硬币例子中的做法相同),并使用它们计算模型参数的后验概率,这类似于在抛硬币例子中计算包含 r 的后验概率。一旦计算了后验概率,就可以使用它对新的奥运会进行预测了。

3.7.1 模型

使用第1章介绍的 k 阶多项式模型和第2章介绍的高斯噪声模型:

$$t_n = w_0 + w_1 x_n + w_2 x_n^2 + \cdots + w_K x_n^K + \epsilon_n$$

其中 $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$ 。其等价的向量表达式为

$$t_n = \mathbf{w}^T \mathbf{x}_n + \epsilon_n$$

其中 $\mathbf{w} = [w_0, \cdots, w_K]^T$, $\mathbf{x}_n = [1, x_n, x_n^2, \cdots, x_n^K]^T$ 。将所有的结果表示成一个矢量 $\mathbf{t} = [t_0, \cdots, t_N]^T$, 并将所有的输入表示成一个简单的矩阵 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]^T$ (类似于式(1-18)), 可以得到整个数据集的表达式:

$$\mathbf{t} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}$$

其中 $\boldsymbol{\epsilon} = [\epsilon_1, \cdots, \epsilon_N]^T$ 。

在这个例子中,假设知道 σ^2 的真实值,这可以使整个问题得以简化。可以应用本章中介绍的所有方法将 σ^2 视为一个随机变量,并得到后验概率分布的分析结果,但是由于计算复杂度较大,可能导致丢失部分主要信息。

将这些变量符号代入贝叶斯规则中,可得:

$$\begin{aligned} p(\mathbf{w} | \mathbf{t}, \mathbf{X}, \sigma^2, \Delta) &= \frac{p(\mathbf{t} | \mathbf{w}, \mathbf{X}, \sigma^2, \Delta) p(\mathbf{w} | \Delta)}{p(\mathbf{t} | \mathbf{X}, \sigma^2, \Delta)} \\ &= \frac{p(\mathbf{t} | \mathbf{w}, \mathbf{X}, \sigma^2) p(\mathbf{w} | \Delta)}{p(\mathbf{t} | \mathbf{X}, \sigma^2, \Delta)} \end{aligned}$$

其中 Δ 对应于定义先验参数 \mathbf{w} 的参数集,这可以使整个模型得到进一步的细化。其图模型如图 3-17 所示。扩展其对应的边缘似然估计为:

$$p(\mathbf{w} | \mathbf{t}, \mathbf{X}, \sigma^2, \Delta) = \frac{p(\mathbf{t} | \mathbf{w}, \mathbf{X}, \sigma^2) p(\mathbf{w} | \Delta)}{\int p(\mathbf{t} | \mathbf{w}, \mathbf{X}, \sigma^2) p(\mathbf{w} | \Delta) d\mathbf{w}} \quad (3-15)$$

使用这个后验概率密度进行预测正是我们所期望的。尤其是,对于新奥运会年份的特征集合

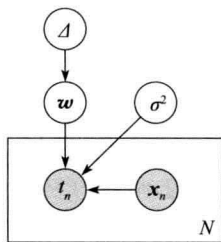


图 3-17 奥运会男子 100 米数据的贝叶斯模型的图模型

\mathbf{x}_{new} , 对应的获胜时间 t_{new} 的密度可按下面公式进行计算:

$$p(t_{\text{new}} | \mathbf{x}_{\text{new}}, \mathbf{X}, t, \sigma^2, \Delta) = \int p(t_{\text{new}} | \mathbf{x}_{\text{new}}, \mathbf{w}, \sigma^2) p(\mathbf{w} | t, \mathbf{X}, \sigma^2, \Delta) d\mathbf{w} \quad (3-16)$$

再次强调等式右边成立的条件为, \mathbf{w} 的后验密度与 \mathbf{w}_{new} 无关, 因此它并没有出现在条件概率中。类似地, 在进行预测时, 并不使用 Δ , 因此它也不出现在 $p(t_{\text{new}} | \mathbf{x}_{\text{new}}, \mathbf{w}, \sigma^2)$ 中。预测结果也可以使用概率的形式表达。例如, 可以计算获胜时间在 9.5 秒以下的概率:

$$p(t_{\text{new}} < 9.5 | \mathbf{x}_{\text{new}}, \mathbf{X}, t, \sigma^2, \Delta) = \int p(t_{\text{new}} < 9.5 | \mathbf{x}_{\text{new}}, \mathbf{w}, \sigma^2) p(\mathbf{w} | t, \mathbf{X}, \sigma^2, \Delta) d\mathbf{w} \quad (3-17)$$

123

3.7.2 似然估计

从前面的章节中可知, 似然估计 $p(t | \mathbf{w}, \mathbf{X}, \sigma^2)$ 是前面章节最大化的那个量。通过模型可以知道:

$$t = \mathbf{X}\mathbf{w} + \varepsilon$$

其中 $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N)$ 。这表示一个高斯随机变量 ε 加上一个常量。正如在 2.7 节所讨论的, 它等于高斯随机变量加上平均数常量。由此可得似然估计为:

$$p(t | \mathbf{w}, \mathbf{X}, \sigma^2) = \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2 \mathbf{I}_N)$$

是一个带有均值 $\mathbf{X}\mathbf{w}$ 和变量 $\sigma^2 \mathbf{I}_N$ 的 N 维高斯密度。这与抛硬币例子中的表达式类似, 都是由式 (3-2) 所给出的二项似然估计。

3.7.3 先验概率

为了生成一个精确后验概念的表达式, 需要选择一个先验概率 $p(\mathbf{w} | \Delta)$, 它是高斯似然估计的共轭。通常, 高斯先验概率是高斯似然估计的共轭。因此, 对 \mathbf{w} 使用高斯先验概率。尤其是,

$$p(\mathbf{w} | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

其中 $\boldsymbol{\mu}_0$ 和 $\boldsymbol{\Sigma}_0$ 将在后面进行详细的讨论。这与硬币例子中的式 (3-4) 类似。从现在起, 表达式中不在显式地取决于 $\boldsymbol{\mu}_0$ 和 $\boldsymbol{\Sigma}_0$, 例如, 为了简单, 将 $p(\mathbf{w} | t, \mathbf{X}, \sigma^2, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ 记为 $p(\mathbf{w} | t, \mathbf{X}, \sigma^2)$ (详见练习 EX 3.10)。

3.7.4 后验概率

本章主要讨论后验概率。与硬币例子相似, 利用我们知道后验概率是高斯分布的事实。这将允许忽略式 (3-15) 中的边缘似然估计, 只处理似然估计和先验直到我们发现与高斯分布成比例的某些量。作为第一步, 可以仅收集与 \mathbf{w} 相关的项而忽略那些与 \mathbf{w} 无关的项:

$$\begin{aligned} p(\mathbf{w} | t, \mathbf{X}, \sigma^2) &\propto p(t | \mathbf{w}, \mathbf{X}, \sigma^2) p(\mathbf{w} | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \\ &= \frac{1}{(2\pi)^{N/2} |\sigma^2 \mathbf{I}|^{1/2}} \exp\left(-\frac{1}{2} (t - \mathbf{X}\mathbf{w})^T (\sigma^2 \mathbf{I})^{-1} (t - \mathbf{X}\mathbf{w})\right) \\ &\quad \times \frac{1}{(2\pi)^{N/2} |\boldsymbol{\Sigma}_0|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{w} - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1} (\mathbf{w} - \boldsymbol{\mu}_0)\right) \\ &\propto \exp\left(-\frac{1}{2\sigma^2} (t - \mathbf{X}\mathbf{w})^T (t - \mathbf{X}\mathbf{w})\right) \exp\left(-\frac{1}{2} (\mathbf{w} - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1} (\mathbf{w} - \boldsymbol{\mu}_0)\right) \\ &= \exp\left\{-\frac{1}{2} \left(\frac{1}{\sigma^2} (t - \mathbf{X}\mathbf{w})^T (t - \mathbf{X}\mathbf{w}) + (\mathbf{w} - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1} (\mathbf{w} - \boldsymbol{\mu}_0)\right)\right\} \end{aligned}$$

124

将括号内的项相乘, 并再一次消除与 \mathbf{w} 不相关的项, 可得,

$$p(\mathbf{w} | \mathbf{t}, \mathbf{X}, \sigma^2) \propto \exp \left\{ -\frac{1}{2} \left(-\frac{2}{\sigma^2} \mathbf{t}^T \mathbf{X} \mathbf{w} + \frac{1}{\sigma^2} \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} + \mathbf{w}^T \boldsymbol{\Sigma}_0^{-1} \mathbf{w} - 2\boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1} \mathbf{w} \right) \right\} \quad (3-18)$$

由于已知后验概率是一个高斯分布，因此可以移除常数项（即不包含 \mathbf{w} 的项）。重新排列多元高斯分布表达式，可以得到与上面类似的表达式：

$$\begin{aligned} p(\mathbf{w} | \mathbf{t}, \mathbf{X}, \sigma^2) &= \mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w) \\ &\propto \exp \left(-\frac{1}{2} (\mathbf{w} - \boldsymbol{\mu}_w)^T \boldsymbol{\Sigma}_w^{-1} (\mathbf{w} - \boldsymbol{\mu}_w) \right) \\ &\propto \exp \left\{ -\frac{1}{2} (\mathbf{w}^T \boldsymbol{\Sigma}_w^{-1} \mathbf{w} - 2\boldsymbol{\mu}_w^T \boldsymbol{\Sigma}_w^{-1} \mathbf{w}) \right\} \end{aligned} \quad (3-19)$$

式(3-18)中 \mathbf{w} 的线性二次项一定等于式(3-19)中的线性二次项。利用这个关系，可以得到 $\boldsymbol{\Sigma}_w$ ：

$$\begin{aligned} \mathbf{w}^T \boldsymbol{\Sigma}_w^{-1} \mathbf{w} &= \frac{1}{\sigma^2} \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} + \mathbf{w}^T \boldsymbol{\Sigma}_0^{-1} \mathbf{w} \\ &= \mathbf{w}^T \left(\frac{1}{\sigma^2} \mathbf{X}^T \mathbf{X} + \boldsymbol{\Sigma}_0^{-1} \right) \mathbf{w} \end{aligned}$$

$$\boldsymbol{\Sigma}_w = \left(\frac{1}{\sigma^2} \mathbf{X}^T \mathbf{X} + \boldsymbol{\Sigma}_0^{-1} \right)^{-1}$$

类似地，通过将式(3-18)与式(3-19)中的线性项相等（并将新的 $\boldsymbol{\Sigma}_w$ 代入到其中），可以得到 $\boldsymbol{\mu}_w$ 的表达式：

$$\begin{aligned} -2\boldsymbol{\mu}_w^T \boldsymbol{\Sigma}_w^{-1} \mathbf{w} &= -\frac{2}{\sigma^2} \mathbf{t}^T \mathbf{X} \mathbf{w} - 2\boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1} \mathbf{w} \\ \boldsymbol{\mu}_w^T \boldsymbol{\Sigma}_w^{-1} \mathbf{w} &= \frac{1}{\sigma^2} \mathbf{t}^T \mathbf{X} \mathbf{w} + \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1} \mathbf{w} \\ \boldsymbol{\mu}_w^T \boldsymbol{\Sigma}_w^{-1} &= \frac{1}{\sigma^2} \mathbf{t}^T \mathbf{X} + \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1} \\ \boldsymbol{\mu}_w^T \boldsymbol{\Sigma}_w^{-1} \boldsymbol{\Sigma}_w &= \left(\frac{1}{\sigma^2} \mathbf{t}^T \mathbf{X} + \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1} \right) \boldsymbol{\Sigma}_w \\ \boldsymbol{\mu}_w^T &= \left(\frac{1}{\sigma^2} \mathbf{t}^T \mathbf{X} + \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1} \right) \boldsymbol{\Sigma}_w \end{aligned}$$

$$\boldsymbol{\mu}_w = \boldsymbol{\Sigma}_w \left(\frac{1}{\sigma^2} \mathbf{X}^T \mathbf{t} + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 \right) \quad (3-20)$$

由对称性可知， $\boldsymbol{\Sigma}_w^T = \boldsymbol{\Sigma}_w$ ，所以有，

$$p(\mathbf{w} | \mathbf{t}, \mathbf{X}, \sigma^2) = \mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w) \quad (3-21)$$

其中

$$\boldsymbol{\Sigma}_w = \left(\frac{1}{\sigma^2} \mathbf{X}^T \mathbf{X} + \boldsymbol{\Sigma}_0^{-1} \right)^{-1} \quad (3-22)$$

$$\boldsymbol{\mu}_w = \boldsymbol{\Sigma}_w \left(\frac{1}{\sigma^2} \mathbf{X}^T \mathbf{t} + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 \right) \quad (3-23)$$

（详见练习 EX 3.12）这些表达式与之前所见过的表达式并没有太大的差异。尤其是，比较式(3-23)与式(1-21)所给出的正则化最小二乘法。事实上，如果 $\boldsymbol{\mu}_0 = [0, 0, \dots, 0]^T$ ，那么这两个表达式几乎是没有任何差别的。鉴于后验概率是高斯分布， \mathbf{w} 最可能的唯一值是后验概率的均值 $\boldsymbol{\mu}_w$ 。这称为 \mathbf{w} 的最大后验概率（MAP）估计，同时也可以认为是联合密度 $p(\mathbf{w}, \mathbf{t} | \mathbf{X}, \sigma^2, \Delta)$ 的最大值（先验概率与似然估计的乘积）。已经认识到第1章

中的平方误差与高斯似然函数相似,在这之后计算最大后验概率值(使用高斯似然函数)与使用正则化最小二乘法等价(参见 EX 3.9)。这一比较有助于建立考虑先验概率影响的直观感觉。

3.7.5 1 阶多项式

因为可以在二维参数空间中可视化密度点,所以我们可使用 1 阶多项式举例说明先验概率和后验概率。输入向量有两个元素, $\mathbf{x}_n = [1, x_n]^T$ 。为了便于可视化,我们重新缩放(标准化)奥运会各个年份的数据:用每一个年份减去第一届奥运会年份(1896 年),然后计算出的数再除以 4。这意味着 x_1 现在是 0, x_2 是 1 等。带有 x 的新的缩放数据将在图 3-18 中给出。

回到游乐场场景,我们分析的第一步是选择先验参数 $\boldsymbol{\mu}_0$ 和 $\boldsymbol{\Sigma}_0$ 。对于 $\boldsymbol{\mu}_0$,我们假设不知道参数应当是多少,并选择 $\boldsymbol{\mu}_0 = [0, 0]^T$ 。对于协方差,我们使用下面的公式:

$$\boldsymbol{\Sigma}_0 = \begin{bmatrix} 100 & 0 \\ 0 & 5 \end{bmatrix}$$

变量 w_0 的值较大,因为在最大似然估计中 w_0 的最优值大于 w_1 的最优值。我们通过设置协方差中非对角线元素为 0,假定先验中这两变量是独立的。这并不妨碍它们在后验概率中是相互依赖的。先验密度的曲线见图 3-19a,根据该模型很难形象地说明它的意义。为了便于理解,在 3-19b 中,表示了由此先验密度获得的若干组参数的相应函数。为了创建这些函数,我们通过 $\boldsymbol{\mu}_0$ 和 $\boldsymbol{\Sigma}_0$ 定义的高斯模型定义对 \mathbf{w} 进行采样,然后代入我们的线性模型 $t_n = w_0 + w_1 x_n$ 。这些例子表明先验密度可以由多种不同模型表示。

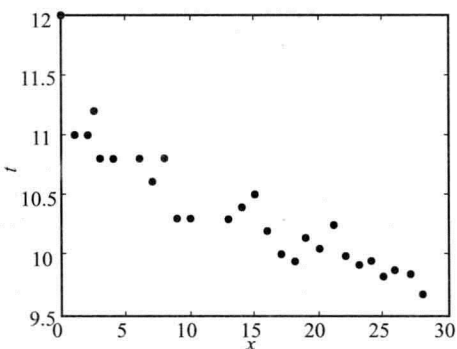
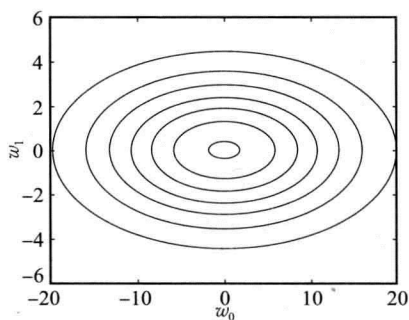
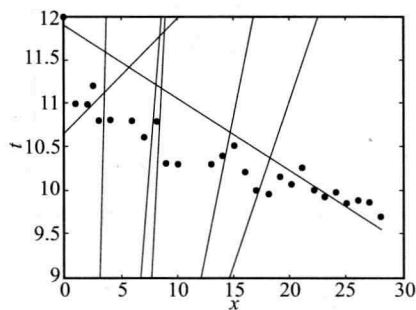


图 3-18 随 x 值变化的奥运会数据



a) 先验密度



b) 由先验参数创建的函数

图 3-19 使用高斯先验计算的奥运会 100 米数据记录 a) 和由先验样本的取值建立的函数 b)

为了说明目的,假定 $\sigma^2 = 10$ (MATLAB 脚本: `olympbayes.m`),当我们观察一个数据点时,可以计算后验分布。使用与第一届奥运会相对应的数据点,数据可以表示为 $\mathbf{x} = [1, 0]^T$ 、 $\mathbf{X} = [1, 0]$ 、 $t = [12]$ 。将带有先验参数的这些值和 $\sigma^2 = 10$ 代入式 (3-21)~式(3-23)中,可以得到如图 3-20a 所示的后验分布。该后验概率中的 w_0 比较确定但 w_1 的信息仍然很少。这也很容易理解——我们已经得到了 $x=0$ 的数据,它包含很多确定截距的信息,但关于确定倾斜度的信息则较少(因为一个数据点不会告诉我们倾斜度)。在图 3-20b 中显示一组由后验参数确定的样本函数。它们与由先验参数获得的样本看起来很不同——尤其是,它们很接近于第一个数据点。

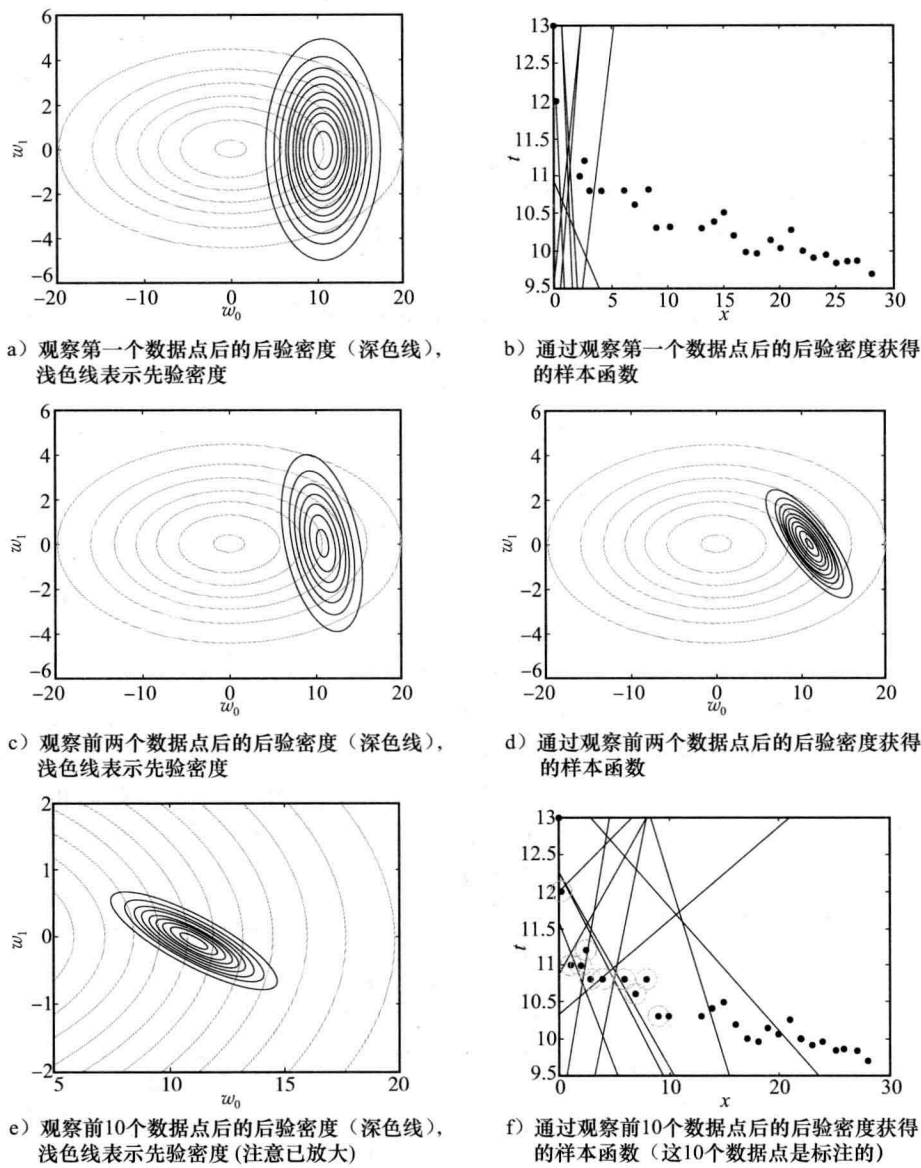


图 3-20 通过增加奥运会观察数据点后的后验密度演变以及通过后验密度获得的样本函数

图 3-20c、d、e 分别表示观察 2、5、10 个数据点后的后验密度的变化。与硬币的例子一样，我们注意到后验变得更加密（ w 值更确定）。而且，随着它的演变，后验变得上翘。这说明这两个参数相互依赖——如果我们增加截距 w_0 就必须减少倾斜率。回顾之前的先验知识，我们假定这两个参数是独立的（ Σ_0 在对角线上有非 0 值），因此这种相互依赖完全来自于数据内的证据。为了有助于可视化后验密度在这个阶段的含义，图 3-20f 显示了一组通过后验密度参数建立的样本函数。与图 3-20b 比较，我们发现后验密度开始依赖于与数据模型相匹配的参数。最终，图 3-21a 展示了所有 27 个数据点均被包含之后的后验概率，并且在图 3-21b 中，表示了所有后验概率对应的函数。这些函数逐渐变得与数据趋势相一致。但仍然有很大的变化，这是由于为了有助于可视化先验密度和后验密度，我们取 $\sigma^2=10$ 的较高值。为了预测，我们可能想使用更多的真实值。在图 3-22a 中，我们显示了在 $\sigma^2=0.05$

(这是我们在 2.7.2 节中得知的最大似然值) 时, 观察全部数据后的后验密度。可以发现在 w 几乎不变的情况下, 后验密度变得更密, 这与之前在图 3-22b 中的函数曲线是一致的。接下来我们将把目光转向预测。

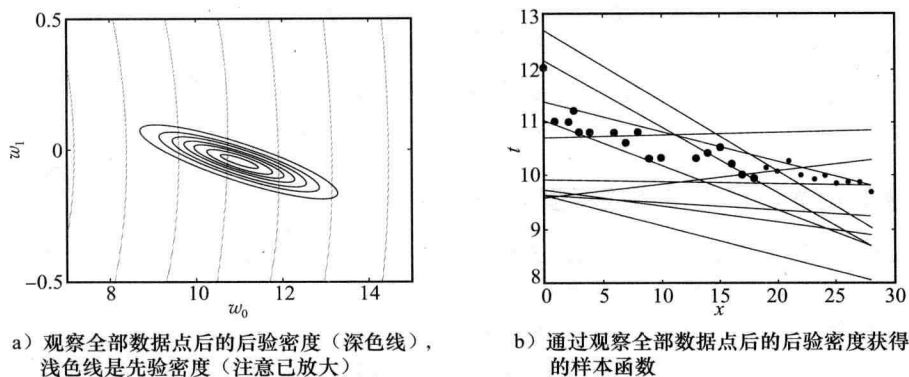


图 3-21 观察全部 27 个奥运会观察数据点后的后验密度样本函数

3.7.6 预测

给一个新的观测值 x_{new} , 我们关注它的密度:

$$p(t_{\text{new}} | x_{\text{new}}, \mathbf{X}, t, \sigma^2)$$

注意, 这并不是在硬币例子中所讲的 w 的条件分布, 我们将通过考虑后验概率的期望积分去掉 w 的期望值 $p(w | t, \mathbf{X}, \sigma^2)$ 。实际上, 我们需要计算:

$$\begin{aligned} p(t_{\text{new}} | x_{\text{new}}, \mathbf{X}, t, \sigma^2) &= \mathbf{E}_{p(w | t, \mathbf{X}, \sigma^2)} \{ p(t_{\text{new}} | x_{\text{new}}, w, \sigma^2) \} \\ &= \int p(t_{\text{new}} | x_{\text{new}}, w, \sigma^2) p(w | t, \mathbf{X}, \sigma^2) dw \end{aligned}$$

这与硬币例子中式 (3-9) 相类似。

我们的模型将 $p(t_{\text{new}} | x_{\text{new}}, w, \sigma^2)$ 看做是 x_{new} 和带有加性高斯噪声的 w 的乘积:

$$p(t_{\text{new}} | x_{\text{new}}, w, \sigma^2) = \mathcal{N}(x_{\text{new}}^T w, \sigma^2).$$

因为这个公式与后验密度公式都是高斯公式, 所以期望的计算结果也是高斯公式。一般地, 如果 $p(w | \mu, \Sigma) = \mathcal{N}(\mu, \Sigma)$, 那么另一个高斯密度期望 $\mathcal{N}(x_{\text{new}}^T w, \sigma^2)$ 为:

$$p(t_{\text{new}} | x_{\text{new}}, \mathbf{X}, t, \sigma^2) = \mathcal{N}(x_{\text{new}}^T \mu_w, \sigma^2 + x_{\text{new}}^T \Sigma_w x_{\text{new}})$$

对于图 3-22a 所示的后验密度, 可以表示为:

$$p(t_{\text{new}} | x_{\text{new}}, \mathbf{X}, t, \sigma^2) = \mathcal{N}(9.5951, 0.0572)$$

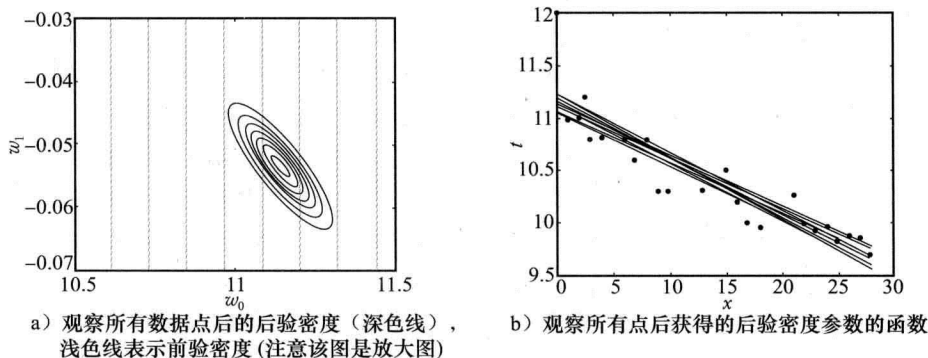


图 3-22 当增加 27 个真实的数据点, 噪声值 $\sigma^2 = 0.05$ 时, 奥运会数据的后验密度 a) 和样本函数 b)

图形曲线如图 3-23 所示。

这个密度看起来像第 2 章用最大似然方法获得的预测密度。然而有一个重要的差异。利用最大似然方法，我们选择似然值最大的模型。为了产生图 3-23 展示的概率密度函数，我们将所有与数据一致的模型和先验（已经对所有后验求平均）求平均。因此，这一密度考虑了已知特定先验和数据情况下 w 中所有的不确定性。

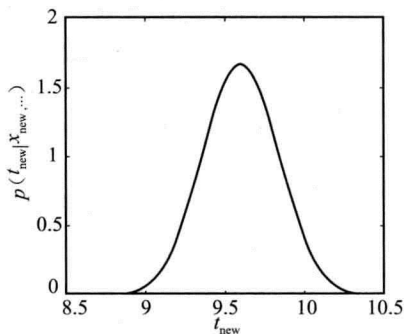


图 3-23 2012 年伦敦奥运会男子 100 米获胜时间的预测分布

3.8 边缘似然估计用于多项式模型阶的选择

在 1.5 节中我们应用交叉验证方法来选择多项式的阶数。交叉验证方法准确地判断出由 3 阶多项式模型产生的数据集。在 3.4 节我们将看到怎样利用边缘概率选择先验密度。现在我们将看到它也能用来选择模型。特别地，我们将利用它来决定利用几阶多项式模型生成数据。

高斯模型的边缘概率函数定义如下：

$$p(t|X, \mu_0, \Sigma_0) = \int p(t|X, w, \sigma^2) p(w | \mu_0, \Sigma_0) dw$$

这与硬币例子中的式 (3-14) 类似。这与以前部分所讨论的预测密度是同样的形式，是高斯模型的另一种形式

$$p(t|X, \mu_0, \Sigma_0) = \mathcal{N}(X\mu_0, \sigma^2 I_N + X\Sigma_0 X^T) \quad (3-24)$$

我们评估 t 值——在训练集上的响应。正如在 1.5 节中，我们从噪声 3 阶多项式模型中生成数据，然后计算 1~7 阶多项式模型的边缘似然。对于每个可能的模型，我们都使用均值为 0 和一致的协方差矩阵计算 w 的高斯先验概率。

例如，对于 1 阶模型

$$\mu_0 = [0, 0]^T, \Sigma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

对于 4 阶模型

$$\mu_0 = [0, 0, 0, 0, 0]^T, \Sigma_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

图 3-14a 显示了数据和真实的多项式 (MATLAB 脚本: margpoly.m)。这个真实的多项式是 $t=5x^3-x^2+x$ ，高斯噪声的均值为 0、方差是 150。1~7 阶模型的边缘似然通过在式 (3-24) 中代入相应的先验密度计算，然后以观测值 t 估计该密度。这个值在图 3-24b 中表示。可以看到边缘似然值在 3 阶模型（真实模型的阶数）处具有最大峰值。利用交叉验证方法的优点在于计算相对容易（不必设置多个不同的数据集）。我们也可以使用所有的数据。然而，正如之前提到的，通常，计算边缘似然非常困难，我们发现借助交叉验证技术往往更容易一些。

边缘似然的计算是以先验密度的参数为条件的，因此改变它们的值常常影响边缘似然和得分最高的模型。为了说明它的影响，可以定义 $\Sigma_0 = \sigma_0^2 I$ 并取不同的 σ_0^2 值。我们已看到 $\sigma_0^2=1$ 的结果。在图 3-25 中看到，当我们降低 σ_0^2 时，高阶模型的形状更好。当我们把 σ_0^2 从 1 减少到

0.3 时,可以发现 7 阶多项式模型是最可能的模型。通过减小 σ_0^2 可以说明参数取值必须越来越小。当参数是 5 时 (回顾 $t=5x^3-x^2+x$), 最合适的模型是 3 阶多项式模型。当我们减少 σ_0^2 时, 这种可能变得越来越小, 具有较小参数值的高阶模型变得更可能。理解我们通过模型所表达的意思非常重要。在这个例子中, 模型包括多项式的阶数和先验的描述, 我们必须认真谨慎地选择合理的先验 (见练习 EX3.11)。

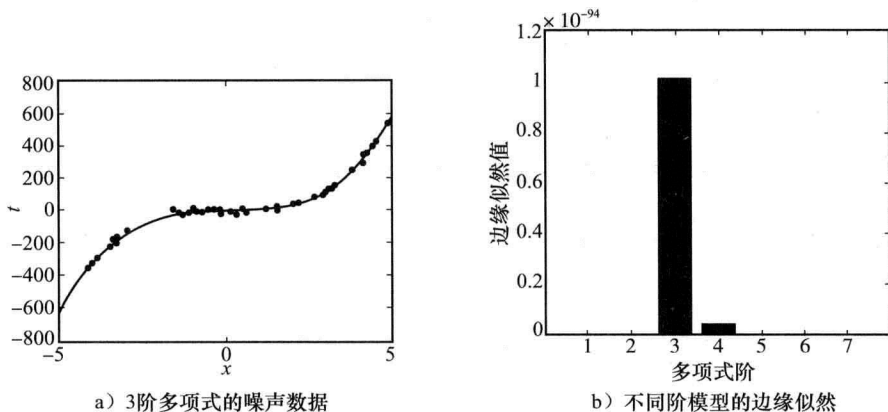


图 3-24 通过公式 $t=5x^3-x^2+x$ 进行数据样本

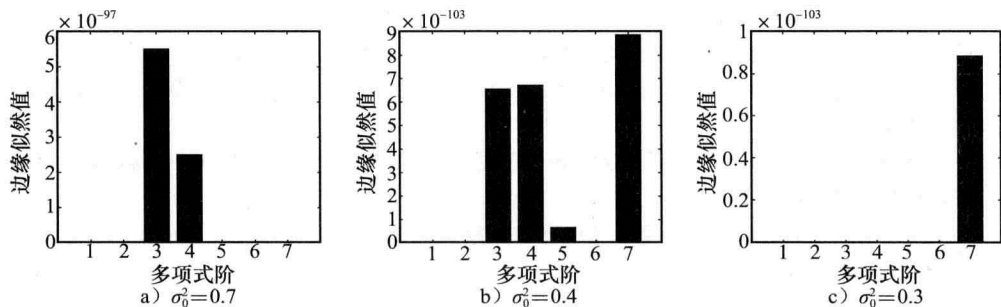


图 3-25 当 $\Sigma_0 = \sigma_0^2 I$ 且 σ_0^2 递减时, 3 阶多项式的边缘似然

3.9 小结

本章主要介绍利用贝叶斯方法完成机器学习任务——把所有参数看做是随机变量。我们用贝叶斯方法分析硬币投掷模型和第 1、2 章中介绍的线性回归模型。在两种情况下, 我们定义了先验密度的参数和似然值, 并计算后验密度。在这两个例子中, 先验密度和似然值是计算后验密度的基础。此外, 我们考虑后验密度的期望值来进行预测并引入边缘概率作为概率模型的选择标准。

不幸的是, 这些表达式往往不易于分析处理, 我们必须求助于采样和近似估计技术。这些技术是现代贝叶斯推理的基础, 并且已经形成一个重要的机器学习研究和开发的领域。第 4 章将主要讨论三个流行技术——点估计、拉普拉斯近似和马尔科夫链—蒙特卡洛法。

3.10 练习

EX 3.1 假定 $\alpha, \beta=1$, β 分布在 $0 \sim 1$ 是均匀的。尤其是, 如果硬币正面着地的概率可以由 r 及关于 r 的 β 先验推出, 那么当参数 $\alpha=1, \beta=1$ 时, r 的先验公式如下所示:

$$p(r) = 1 \quad (0 \leq r \leq 1)$$

利用该先验信息, 计算在 N 次掷币中出现 y 次正面的后验密度 (例如, 将该先验公式与二项分布的概率相乘, 获得类似 β 密度的结果)。

EX 3.2 对于下面的先验公式, 重复上一题的练习, β 概率密度函数的特殊形式为:

$$p(r) = \begin{cases} 2r & 0 \leq r \leq 1 \\ 0 & \text{其他} \end{cases}$$

当 $p(r)=2r$ 时, 先验参数 α 、 β 的值是多少?

EX 3.3 对于下面的先验公式, 重复上一题的练习, β 概率密度函数的特殊形式为:

$$p(r) = \begin{cases} 3r^2 & 0 \leq r \leq 1 \\ 0 & \text{其他} \end{cases}$$

先验参数是什么?

EX 3.4 对于前三题, 有效的先验样本数 (α 和 β) 是多少 (例如, 当它们相等时出现了多少次正面向上和正面向下)?

EX 3.5 如果一个随机变量 R 服从 β 分布,

$$p(r) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1} (1-r)^{\beta-1}$$

推导出 r 期望值的表达式 $\mathbf{E}_{p(r)}\{r\}$, 使用下面的 gamma 函数:

$$\Gamma(n+1) = n\Gamma(n)$$

提示: 使用下面的公式, $\int_{r=0}^1 r^{a-1} (1-r)^{b-1} dr = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$

EX 3.6 使用前一练习的结果和下面的等式:

$$\text{var}\{r\} = \mathbf{E}_{p(r)}\{r^2\} - (\mathbf{E}_{p(r)}\{r\})^2$$

推导出 $\text{var}\{r\}$ 的表达式, 使用上题中的 gamma 函数。

EX 3.7 以不同的货摊观察 20 次投掷, 其中 9 次正面朝上, 计算三种情况的后验密度, 每种情况下赢的概率和边缘似然。

EX 3.8 使用 Matlab 模拟掷币, 正面朝上概率是 0.7。掷币 100 次, 计算三种情况的后验密度, 每种情况下赢的概率和边缘似然。

EX 3.9 在 3.7.4 节中, 我们获得了奥运会 100 米数据线性模型的高斯后验, 代入 $\mu_0 = [0, 0, \dots, 0]^T$, 我们看到后验密度的均值

$$\mu_w = \frac{1}{\sigma^2} \left(\frac{1}{\sigma^2} \mathbf{X}^T \mathbf{X} + \Sigma_0^{-1} \right)^{-1} \mathbf{X}^T \mathbf{t}$$

与最小二乘法之间的相似性

$$\hat{\mathbf{w}} = \left(\mathbf{X}^T \mathbf{X} + N\lambda \mathbf{I} \right)^{-1} \mathbf{X}^T \mathbf{t}$$

根据这一特例, 计算先验协方差矩阵 Σ_0 , 使得两者相同。换句话说, 根据 λ 找到 Σ_0 。

EX 3.10 重新绘制以 μ_0 和 Σ_0 为条件的奥运会 100 米数据的先验密度 w 的图形。

EX 3.11 通过图 3-25, 我们分析了减少 σ_0^2 对边缘似然的影响。使用 Matlab 分析, 增加 σ_0^2 对边缘似然的影响。

EX 3.12 当对奥运会数据进行贝叶斯分析时, 假定 σ^2 是已知的。反之, 假定 w 是已知的, 并且先验密度关于 σ^2 的 gamma 变换如下:

$$p(\sigma^2 | \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} (\sigma^2)^{-\alpha-1} \exp\left\{-\frac{\beta}{\sigma^2}\right\}$$

后验密度也可以做类似的 gamma 变换。推导出后验密度的参数。

其他阅读材料

- [1] Ben Calderhead and Mark Girolami. Estimating Bayes factors via thermodynamic integration and population mcmc. *Comput. Stat. Data Anal.*, 53:4028–4045, October 2009.

文章主要描述计算贝叶斯模型中不易解析的边缘似然值的新方法。

- [2] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, second edition, 2004.

最受欢迎的贝叶斯推理手册之一，书中详细介绍了贝叶斯推理的实用技巧。

- [3] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision*, pages 343–356, Springer, 1996.

将贝叶斯方法应用到人机交互领域的有趣示例。作者利用采样技术推理用户手势使用的后验概率。

- [4] Michael Jordan, editor. *Learning in Graphical Models*. MIT Press, 1999.

关于图模型领域知识的介绍以及如何利用图模型完成学习任务。

- [5] Christian Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer, second edition, 2007.

- [6] Tian-Rui Xu et al. Inferring signaling pathway topologies from multiple perturbation measurement of specific biochemical species. *Science Signalling*, 3(113), 2010.

这篇文章主要讲述如何利用边缘似然值选择贝叶斯模型去解决生物学领域的科学问题。这也是大规模贝叶斯采样的有趣示例。

贝叶斯推理

第3章介绍了把贝叶斯方法应用到机器学习所需要的主要概念。在贝叶斯框架中，所有未知量都视为随机变量。用分布来描述每个参数而不是用值。参数估计的不确定性自然地会影响我们做的任何预测。先前看到先验和似然相结合的两个例子是共轭的，这意味着后验和先验有相同的形式，可以用解析方法进行计算。我们可以证明选择共轭先验和似然的组合是罕见的。其余部分，不能计算后验，必须采取近似的方法。本章将介绍三种这样的近似技术。

4.1 非共轭模型

在第3章中，我们看到用两个模型进行精确贝叶斯推理是可能的。在第一种情况下，建立了一个掷硬币和一个 β 先验与二项似然相结合的模型，这意味着我们可以说明后验也属于 β 家族。在第二个例子中，高斯先验加上高斯似然导致一个高斯后验。事实上，我们知道后验的形式意味着我们并不需要计算归一化常数（分母，例如，式（3-3））。只要能找到与感兴趣密度相成比例的分布形式（例如，一个 β 或高斯分布），我们就能确定归一化要考虑其本身。 β 二项分布和高斯-高斯组合不是唯一可以使用的共轭先验-似然对。其他两个典型的例子分别是离散和连续数据的多项-狄利克雷（multinomial-Dirichlet）和 γ -高斯。

对于许多模型，选择共轭先验和似然是不可能的（或者从建模的观点是不可行的），我们不得不使用近似方法。在本章中，通过一个二值分类问题来介绍三种近似技术。二值分类是机器学习中常见的问题，并不存在共轭先验和似然的结合。我们将着眼于三种技术：点估计、近似密度和采样，这三种技术在机器学习中被广泛使用。

4.2 二值响应

图4-1显示了一个与以前看到的有所不同的数据集，每个对象都由两个属性（ x_1 和 x_2 ）来描述，还有一个二值响应 $t=\{0, 1\}$ 。根据对象的响应为其绘制一个符号：如果 $t=0$ ，点被绘成一个圆圈；如果 $t=1$ ，则绘成一个正方形。将使用这些数据建立一个模型，使我们能够预测一个新对象的响应（0或1，圆圈或正方形）。这个任务被视为分类——我们希望能够把对象划分为类别组中的一类（在这种情况下有两类）。分类是机器学习的主要问题之一，我们将在第5章介绍几种其他的分类算法。

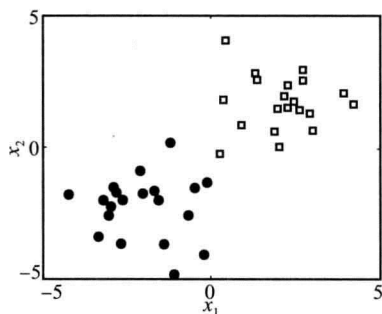


图4-1 二值响应的一个例子，每个对象都由 x_1 和 x_2 两个属性，以及一个二值目标 $t=\{0, 1\}$ 定义。 $t=0$ 的点被绘成一个圆圈， $t=1$ 的点绘成一个正方形

4.2.1 二值响应模型

使用下面的向量和矩阵表示我们的数据：

$$\mathbf{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}$$

这个模型（带有参数 \boldsymbol{w} ）允许我们为一些新的观察对象 $\boldsymbol{x}_{\text{new}}$ 预测 t_{new} 。

正如3.7节中奥运会的例子，我们需要计算基于模型参数的后验密度。根据贝叶斯规则得出：

140

$$p(\boldsymbol{w} | \boldsymbol{t}, \boldsymbol{X}) = \frac{p(\boldsymbol{t} | \boldsymbol{X}, \boldsymbol{w}) p(\boldsymbol{w})}{p(\boldsymbol{t} | \boldsymbol{X})} \quad (4-1)$$

其中，边际似然函数 $p(\boldsymbol{t} | \boldsymbol{X})$ 为：

$$p(\boldsymbol{t} | \boldsymbol{X}) = \int p(\boldsymbol{t} | \boldsymbol{X}, \boldsymbol{w}) p(\boldsymbol{w}) d\boldsymbol{w}$$

先验：我们为 $p(\boldsymbol{w})$ 使用一个高斯密度。特别地， $p(\boldsymbol{w}) = \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ 。为了保持一致性，假设 $p(\boldsymbol{w})$ 依赖于 σ^2 ，将先验表示为 $p(\boldsymbol{w} | \sigma^2)$ 。在前面的章节中，选择高斯密度往往出于分析的方便。既然本章不能够依靠共轭，我们就没有限制先验密度的选择。然而，本章关注的是克服非共轭的方法，高斯就足够了。建议读者尝试做有关本章介绍的具有不同形式的先验密度 $p(\boldsymbol{w})$ 的练习题。

似然：为了使似然 $p(\boldsymbol{t} | \boldsymbol{X}, \boldsymbol{w})$ 有所进展，我们从假设 \boldsymbol{t} 的元素是相互条件独立的开始（参见2.7.1节），有条件依赖于 \boldsymbol{w} ：

$$p(\boldsymbol{t} | \boldsymbol{X}, \boldsymbol{w}) = \prod_{n=1}^N p(t_n | \boldsymbol{x}_n, \boldsymbol{w})$$

t_n 是一个二值变量，表示第 n 个对象 \boldsymbol{x}_n 的类别（0或1）。在前面章节的高斯奥运会例子中，我们视 t_n 为均值 $\boldsymbol{w}^\top \boldsymbol{x}_n$ 和方差 σ^2 的高斯随机变量，但它只适用于实值 t_n 。相反，可以把 t_n 建模为一个二值随机变量——对每个 n 都是一次单独的抛硬币问题。而不是均值和方差，这个随机变量由类别为1的概率来刻画（类别为0的概率等于1减去类别为1的概率）。为了避免混乱，用 T_n 表示这个随机变量（为了从实例中区分我们所观察到的 t_n ）。因此，可以把每个 n 当做一个概率事件：

$$p(\boldsymbol{t} | \boldsymbol{X}, \boldsymbol{w}) = \prod_{n=1}^N p(T_n = t_n | \boldsymbol{x}_n, \boldsymbol{w}) \quad (4-2)$$

当观察类别1时，这个似然函数给类别1赋予很高的概率；同样，当观察类别0时，类别0的概率也很高。当所有训练点都被很好地预测时，它将达到最大值1。

现在，我们的任务是选择一个产生概率的有关 \boldsymbol{x}_n 和 \boldsymbol{w} 的函数 $f(\boldsymbol{x}_n; \boldsymbol{w})$ 。通常的方法是采用一个简单的线性函数（如 $f(\boldsymbol{x}_n; \boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{x}_n$ ），然后通过一个二级函数压缩其输出，产生一个结果来确保它产生一个有效的概率。这样一个压缩函数是sigmoid函数，如图4-2所示。当 $\boldsymbol{w}^\top \boldsymbol{x}$ 增加时，值收敛到1；当它减小时，值收敛到0。sigmoid函数定义为：

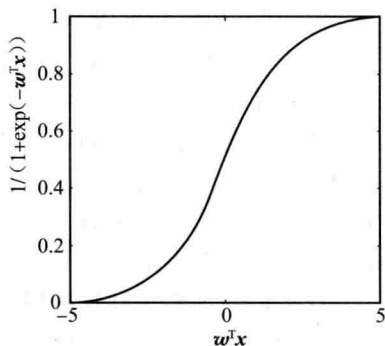


图4-2 压缩实值总在到0~1的sigmoid函数

$$P(T_n = 1 | \boldsymbol{x}_n, \boldsymbol{w}) = \frac{1}{1 + \exp(-\boldsymbol{w}^\top \boldsymbol{x}_n)} \quad (4-3)$$

这个表达式给出了 $T_n = 1$ 时的概率。在我们的似然中，需要实际观测的概率，其中一些为0。因为 T_n 只能取0或1，所以可以很容易地使用式(2-2)计算 $P(T_n = 0 | \boldsymbol{x}, \boldsymbol{w})$ ：

$$\begin{aligned} P(T_n = 0 | \boldsymbol{x}_n, \boldsymbol{w}) &= 1 - P(T_n = 1 | \boldsymbol{x}_n, \boldsymbol{w}) \\ &= 1 - \frac{1}{1 + \exp(-\boldsymbol{w}^\top \boldsymbol{x}_n)} \end{aligned}$$

$$= \frac{\exp(-\mathbf{w}^T \mathbf{x}_n)}{1 + \exp(-\mathbf{w}^T \mathbf{x}_n)} \quad (4-4)$$

结合式 (4-3) 和式 (4-4) 为 $P(T_n = t_n | \mathbf{x}_n, \mathbf{w})$ 产生一个单一表达式:

$$P(T_n = t_n | \mathbf{x}_n, \mathbf{w}) = P(T_n = 1 | \mathbf{x}_n, \mathbf{w})^{t_n} P(T_n = 0 | \mathbf{x}_n, \mathbf{w})^{1-t_n}$$

其中, 观测数据 (t_n) 导致与它相关的项起作用, 不相关的项不起作用。

将它代入式 (4-2) 可以得到所有 n 个训练点的似然:

$$\begin{aligned} p(t | \mathbf{X}, \mathbf{w}) &= \prod_{n=1}^N P(T_n = 1 | \mathbf{x}_n, \mathbf{w})^{t_n} P(T_n = 0 | \mathbf{x}_n, \mathbf{w})^{1-t_n} \\ &= \prod_{n=1}^N \left(\frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_n)} \right)^{t_n} \left(\frac{\exp(-\mathbf{w}^T \mathbf{x}_n)}{1 + \exp(-\mathbf{w}^T \mathbf{x}_n)} \right)^{1-t_n} \end{aligned} \quad (4-5)$$

后验: 将似然的定义与先前选择的高斯先验相结合是必要的, 理论上, 为了计算后验密度 $p(\mathbf{w} | \mathbf{X}, t, \sigma^2)$ 。一旦有了后验密度, 我们就能通过相对于这个密度的期望来预测新对象的响应 (类):

$$P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, t) = \mathbf{E}_{p(\mathbf{w} | \mathbf{X}, t, \sigma^2)} \left\{ \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_{\text{new}})} \right\}$$

实际上, 这并不简单。后验没有任何标准形式。为了评价特定 \mathbf{w} 的后验, 需要评价式 (4-1) 的分子和分母。分子是好的——可以评估 \mathbf{w} 的高斯先验密度和我们刚刚定义的似然, 并将两者相乘。分母则是一个问题, 我们不能用解析方法计算综合需求来计算边际似然:

$$Z^{-1} = p(t | \mathbf{X}, \sigma^2) = \int p(t | \mathbf{X}, \mathbf{w}) p(\mathbf{w} | \sigma^2) d\mathbf{w}$$

换句话说, 有一个函数 $g(\mathbf{w}; \mathbf{X}, t, \sigma^2) = p(t | \mathbf{X}, \mathbf{w}) p(\mathbf{w} | \sigma^2)$, 它与后验 $p(\mathbf{w} | \mathbf{X}, t, \sigma^2) = Z^{-1} g(\mathbf{w}; \mathbf{X}, t, \sigma^2)$ 成正比, 但是我们不知道概率的一致性 Z^{-1} (注意, 这个常数通常定义为 Z^{-1} 而不是 Z)。我们只剩下 3 个选项:

- 1) 找到与最高后验值一致的 \mathbf{w} 单值。由于 $g(\mathbf{w}; \mathbf{X}, t, \sigma^2)$ 与后验成正比, 所以 $g(\mathbf{w}; \mathbf{X}, t, \sigma^2)$ 的最大值也将与后验的最大值一致。 Z^{-1} 不是 \mathbf{w} 的函数。
- 2) 用其他一些可以分析计算的密度来近似 $p(\mathbf{w} | \mathbf{X}, t, \sigma^2)$ 。
- 3) 只知道 $g(\mathbf{w}; \mathbf{X}, t, \sigma^2)$ 的情况下, 直接从后验 $p(\mathbf{w} | \mathbf{X}, t, \sigma^2)$ 进行采样。

第一个选项绝不是贝叶斯——我们将不得不基于 \mathbf{w} 的单一值而不是密度对新对象做出预测。然而, 这很容易做到, 这使得它成为流行的技术。第二个选项给我们留下了一个容易处理的密度 (我们可以选择任何我们喜欢的密度), 但如果选择的密度非常不同于后验, 那么模型将非常不可靠。最后一个选项允许我们从后验中采样 (因此对我们可能需要的任何期望都能获得良好的近似), 但可能会很困难。

在任何不能直接计算后验密度的问题上, 这 3 个选项都是可用的。所有这 3 个选项有好有坏, 选择哪一个依赖于所处理问题的特殊性 (计算限制)。现在, 我们将依次描述。

4.3 点估计: 最大后验估计方案

4.2 节表明, 当不能计算后验密度 $p(\mathbf{w} | \mathbf{X}, t, \sigma^2)$ 时, 我们可以计算一些与其成正比的量 $g(\mathbf{w}; \mathbf{X}, t, \sigma^2)$ 。这等价于先验和似然的乘积。最小化 $g(\mathbf{w}; \mathbf{X}, t, \sigma^2)$ 的 \mathbf{w} 值与后验最大时的 \mathbf{w} 值一致。这将是单一的最有可能是 $\hat{\mathbf{w}}$ 值 (在后验下), 如果我们决定使用点估计, 则这是一个明智的选择。第 2 章讲解了寻找最大似然时的 $\hat{\mathbf{w}}$ 值。这里的想法非常类似, 除了我们现在最大化似然和先验的乘积。该解决方案是在 3.7.4 节第一次看到的最大后验估计 (MAP), 在机器学习中很普遍。

与寻找最大似然的解决方案一样, 很容易找到 \mathbf{w} 值使得 $\log g(\mathbf{w}; \mathbf{X}, t)$ 最大而不是

$g(\mathbf{w}; \mathbf{X}, t)$ 最大:

$$\log g(\mathbf{w}; \mathbf{X}, t) = \log p(t | \mathbf{X}, \mathbf{w}) + \log p(\mathbf{w} | \sigma^2)$$

与线性模型的最大似然解决方案不同, 我们无法通过对表达式进行微分并使其为 0 来为 \mathbf{w} 获得一个精确的表达式。相反, 我们可以使用许多优化算法中的任何一种从给 \mathbf{w} 一个猜想值开始, 然后不断地更新, 以这种方式使 $g(\mathbf{w}; \mathbf{X}, t)$ 不断增加直到最大。牛顿-拉夫森过程 (见注解 4.1) 就是使用式 (4-6) 不断更新 \mathbf{w} 值的一种方法:

$$\mathbf{w}' = \mathbf{w} - \left(\frac{\partial^2 \log g(\mathbf{w}; \mathbf{X}, t)}{\partial \mathbf{w} \partial \mathbf{w}^\top} \right)^{-1} \frac{\partial \log g(\mathbf{w}; \mathbf{X}, t)}{\partial \mathbf{w}} \quad (4-6)$$

注解 4.1 (Newton-Raphson 方法): Newton-Raphson 方法 (又称为牛顿方法) 是寻找函数为 0 的点的一般方法。例如, 寻找当函数 $f(x)=0$ 时的点。假设一个当前 0 点估计 x_n , 通过移动函数在 x_n 处的切线经过 x 轴的点来更新它。这个点可以通过估计 $f(x)$ 的变化量除以 x 的变化量这样一个梯度得到。定义 $\partial f(x)/\partial x$ 为 $f'(x)$:

$$\begin{aligned} f'(x_n) &= \frac{f(x_n) - 0}{x_n - x_{n+1}} \\ (x_n - x_{n+1}) f'(x_n) &= f(x_n) \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \end{aligned}$$

这个方法也能用来寻找最小值和最大值, 因为这些都是倾斜度通过 0 的点。因此, 我们用 $f(x)$ 的导数 $f'(x)$ 代替 $f(x)$, 用 $f'(x)$ 的导数 $f''(x)$ 代替 $f'(x)$:

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

这很容易扩展到向量函数, 比如 \mathbf{x} 。在这种情况下, $f'(x_n)$ 由关于 \mathbf{x}_n 的偏导数向量代替, $1/f''(x)$ 由关于 \mathbf{x}_n 的 Hessian 矩阵 (见注解 2.6) 的逆 $\partial^2 f(\mathbf{x})/\partial \mathbf{x} \partial \mathbf{x}^\top$ 代替——在 $\mathbf{x}=\mathbf{x}_n$ 处评估。

\mathbf{w} 的新版本 (\mathbf{w}') 由 \mathbf{w} 减去 Hessian 的逆 (见注解 2.6) 与偏导数向量的乘积。对于 \mathbf{w} 的任何初始值, 这个迭代过程将更新 \mathbf{w} 直到其成为倾斜度为 0 的点。为了检测我们得到的点收敛于最大的点, 可以检测 Hessian 矩阵以确保它是负定的, 就像 2.7.3 节中我们为最大似然所做的一样。

为了计算 1 阶导数向量, 我们首先用式 (4-2) 和式 (4-5) 为 $\log g(\mathbf{w}; \mathbf{X}, t)$ 扩展表达式:

$$\begin{aligned} \log g(\mathbf{w}; \mathbf{X}, t) &= \sum_{n=1}^N \log P(T_n = t_n | \mathbf{x}_n, \mathbf{w}) + \log p(\mathbf{w} | \sigma^2) \\ &= \sum_{n=1}^N \log \left(\frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x}_n)} \right)^{t_n} \left(\frac{\exp(-\mathbf{w}^\top \mathbf{x}_n)}{1 + \exp(-\mathbf{w}^\top \mathbf{x}_n)} \right)^{1-t_n} + \log p(\mathbf{w} | \sigma^2) \end{aligned}$$

为了防止表达式一开始就太复杂, 我们使用以下的简记:

$$P_n = P(T_n = 1 | \mathbf{w}, \mathbf{x}_n) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x}_n)}$$

因此, 假设 \mathbf{w} 是 D 维的, 我们有以下表达式:

$$\begin{aligned} \log g(\mathbf{w}; \mathbf{X}, t) &= \log p(\mathbf{w} | \sigma^2) + \sum_{n=1}^N \log P_n^{t_n} + \log(1 - P_n)^{1-t_n} \\ &= -\frac{D}{2} \log 2\pi - D \log \sigma - \frac{1}{2\sigma^2} \mathbf{w}^\top \mathbf{w} + \sum_{n=1}^N t_n \log P_n + (1 - t_n) \log(1 - P_n) \end{aligned}$$

其中, 前三项是 (高斯) 先验的对数。为了寻找偏导数向量, 我们对 P_n 的偏导数, 使用链式规则 (见注解 4.2) 给出表达式:

$$\begin{aligned}
\frac{\partial \log g(\mathbf{w}; \mathbf{X}, t)}{\partial \mathbf{w}} &= -\frac{1}{\sigma^2} \mathbf{w} + \sum_{n=1}^N \left(\frac{t_n}{P_n} \frac{\partial P_n}{\partial \mathbf{w}} + \frac{1-t_n}{1-P_n} \frac{\partial (1-P_n)}{\partial \mathbf{w}} \right) \\
&= -\frac{1}{\sigma^2} \mathbf{w} + \sum_{n=1}^N \left(\frac{t_n}{P_n} \frac{\partial P_n}{\partial \mathbf{w}} + \frac{1-t_n}{1-P_n} \frac{\partial P_n}{\partial \mathbf{w}} \right) \quad (4-7)
\end{aligned}$$

注解 4.2 (链式规则): 当要获得偏导数时, 通常使用链式规则很方便。链式规则描述如下:

$$\frac{\partial f(g(\mathbf{w}))}{\partial \mathbf{w}} = \frac{\partial f(g(\mathbf{w}))}{\partial g(\mathbf{w})} \frac{\partial g(\mathbf{w})}{\partial \mathbf{w}}$$

作为一个例子, 让

$$f(\mathbf{w}) = t_n \log P_n$$

其中

$$p_n = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_n)}$$

为了计算 $\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}}$, 我们按照下面的公式使用链式规则:

$$\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial f(\mathbf{w})}{\partial P_n} \frac{\partial P_n}{\partial \mathbf{w}} = \frac{t_n}{P_n} \frac{\partial P_n}{\partial \mathbf{w}}$$

其中, 再次使用链式规则把 $\frac{\partial (1-P_n)}{\partial \mathbf{w}}$ 变成 $-\frac{\partial P_n}{\partial \mathbf{w}}$:

$$\frac{\partial (1-P_n)}{\partial \mathbf{w}} = \frac{\partial (1-P_n)}{\partial P_n} \frac{\partial P_n}{\partial \mathbf{w}} = -\frac{\partial P_n}{\partial \mathbf{w}}$$

为了计算 $\frac{\partial P_n}{\partial \mathbf{w}}$, 还可以再次使用链式规则:

$$\begin{aligned}
\frac{\partial P_n}{\partial \mathbf{w}} &= \frac{\partial (1 + \exp(-\mathbf{w}^T \mathbf{x}_n))^{-1}}{\partial (1 + \exp(-\mathbf{w}^T \mathbf{x}_n))} \frac{\partial (1 + \exp(-\mathbf{w}^T \mathbf{x}_n))}{\partial \mathbf{w}} \\
&= -\frac{1}{(1 + \exp(-\mathbf{w}^T \mathbf{x}_n))^2} \exp(-\mathbf{w}^T \mathbf{x}_n) (-\mathbf{x}_n) \\
&= \frac{\exp(-\mathbf{w}^T \mathbf{x}_n)}{(1 + \exp(-\mathbf{w}^T \mathbf{x}_n))^2} \mathbf{x}_n \\
&= \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_n)} \frac{\exp(-\mathbf{w}^T \mathbf{x}_n)}{1 + \exp(-\mathbf{w}^T \mathbf{x}_n)} \mathbf{x}_n \\
&= P_n (1 - P_n) \mathbf{x}_n \quad (4-8)
\end{aligned}$$

把式 (4-8) 代入式 (4-7), 就得到需要的偏导数向量:

$$\begin{aligned}
\frac{\partial \log g(\mathbf{w}; \mathbf{X}, t)}{\partial \mathbf{w}} &= -\frac{1}{\sigma^2} \mathbf{w} + \sum_{n=1}^N (\mathbf{x}_n t_n (1 - P_n) - \mathbf{x}_n (1 - t_n) P_n) \\
&= -\frac{1}{\sigma^2} \mathbf{w} + \sum_{n=1}^N \mathbf{x}_n (t_n - t_n P_n - P_n + t_n P_n) \\
&= -\frac{1}{\sigma^2} \mathbf{w} + \sum_{n=1}^N \mathbf{x}_n (t_n - P_n) \quad (4-9)
\end{aligned}$$

为了计算 2 阶导数的 Hessian 矩阵, 我们对 \mathbf{w}^T 再次求微分。注意 $\frac{\partial P_n}{\partial \mathbf{w}^T} = \left(\frac{\partial P_n}{\partial \mathbf{w}} \right)^T$, 我们得到以下表达式:

$$\frac{\partial^2 \log g(\mathbf{w}; \mathbf{X}, t)}{\partial \mathbf{w} \partial \mathbf{w}^T} = -\frac{1}{\sigma^2} \mathbf{I} - \sum_{n=1}^N \mathbf{x}_n \frac{\partial P_n}{\partial \mathbf{w}^T} = -\frac{1}{\sigma^2} \mathbf{I} - \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^T P_n (1 - P_n) \quad (4-10)$$

Hessian 矩阵中需要注意的一点是, 因为 $0 \leq P_n \leq 1$, 所以它对于任何 x_n 和任何 w 都是负定的 (见 2.7.3 节)。因此, 仅有一个最优解且为最大值。无论 w 的值是什么, 牛顿-拉弗森过程必须收敛到与后验密度最大值一致。这是选择先验和似然函数的结果, 改变两者之一都可能加大后验密度优化的难度。

现在我们已经具备了执行牛顿-拉弗森过程的一切准备, 并找到了 w 潜在的最优值。从 $w = [0, 0]^T$ 开始, 令 $\sigma^2 = 10$, 仅 9 次迭代后 (MATLAB 脚本: logmap.m) 过程就收敛 (w 的变化不是很重要)。这个时期 w 的 2 个元素的变化如图 4-3 所示。接着前面的章节, 我们把 w 的最大值叫做 \hat{w} 。

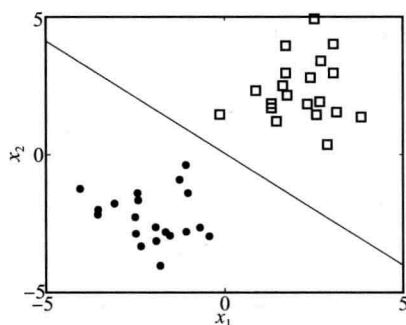
使用 \hat{w} , 我们可以计算任何 x 响应为 1 的概率。尤其是, 我们观察一个新 x_{new} 、一组新属性, 它响应可能为 1 的概率由以下公式计算得到:

$$P(T_{\text{new}} = 1 | x_{\text{new}}, \hat{w}) = \frac{1}{1 + \exp(-\hat{w}^T x_{\text{new}})} \quad (4-11)$$

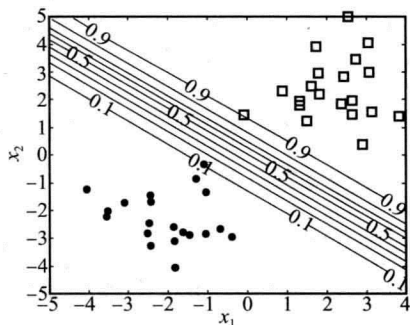
已知这个新对象的两个可能的响应 (类), 如果概率大于 0.5 则划分为正方形类 ($T_{\text{new}} = 1$); 否则, 划分为圆形类 ($T_{\text{new}} = 0$)。在这种情况下, 符合 $P(T=1 | x, \hat{w}) = 0.5$ 的 x 将形成一条直线, 我们认为是**决策边界**——在直线一侧的点属于一类, 另一侧的点属于另一类。为了画出决策边界, 我们利用 $P(T=1 | x, \hat{w}) = 0.5$ 暗示着 $\hat{w}^T x = 0$ 的事实 (见练习 EX 4.5)。如果我们扩展这个表达式, 我们就可以得到决策边界作为 x_1 和 x_2 的函数:

$$\begin{aligned} 0 &= \hat{w}^T x = \hat{w}_1 x_1 + \hat{w}_2 x_2 \\ \hat{w}_2 x_2 &= -\hat{w}_1 x_1 \\ x_2 &= -\frac{\hat{w}_1 x_1}{\hat{w}_2} \end{aligned}$$

如图 4-4a 所示。如果我们想用一条直线划分两个类别, 这看起来是一个相当合理的选择。在图 4-4b 中, 我们画出了 $P(T=1 | x, \hat{w})$ 作为 x 函数的曲线 (MATLAB 脚本: logmap.m)。接近正方形的概率为 1 (正方形是 $t_n=1$ 的对象), 接近圆形的概率为 0。两者之间数据的概率为 0.5 左右, 反映了对象可能与两组等距这样一个事实。



a) $P(T=1|x, \hat{w})=0.5$ 时的数据和直线。当把 0.5 作为阈值时, 将位于直线之上的新点分为正方形, 下面的点分为圆形



b) 等概率线显示了 $P(T=1|x, \hat{w})$ 为 x 的函数, 表示将新对象分类为正方形的概率

图 4-4 二值响应例子中的推理函数

这个优化的结果是我们有了一个可以预测的模型。这个模型是基于参数 \hat{w} 的点估计,

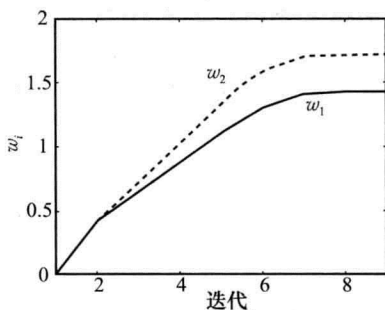


图 4-3 通过牛顿-拉弗森方法找到的对应于后验密度最大值的 w , 图示为 w 成员的演变过程

参数是通过寻找与最大后验 $p(\mathbf{w} | \mathbf{X}, \mathbf{t}, \sigma^2)$ 一致的 \mathbf{w} 值。最大后验估计解决方案在机器学习中是常见的，因为这种方法很容易找到 $\hat{\mathbf{w}}$ 。对于任何先验和似然的组合都可以按照上面描述的步骤得到最优值。在某些问题中，优化过程不一定和上面一致，后验可能有多个最大值（可能甚至某些最小值）。很难知道我们通过牛顿-拉弗森找到的最大值是全局最优。

第3章已经发现计算向量 \mathbf{w} 的密度函数而不是仅仅考虑点估计的优势。采用这样的思想，当我们无法通过计算准确找到一个近似 $p(\mathbf{w} | \mathbf{X}, \mathbf{t}, \sigma^2)$ 的后验密度时，我们将转移到第二个选项。

148

4.4 拉普拉斯近似

在机器学习中有多种近似方法，用较容易处理的近似来代替棘手的后验密度。最流行的就是拉普拉斯近似^①。主要思想是用高斯近似感兴趣的密度。由于我们可以方便地操纵高斯，所以这看起来是合理的选择——用给出的高斯后验很容易计算需要预测的期望。然而，我们应该始终牢记我们的预测仅仅只是逼近的。如果真实的后验不是高斯，那么我们的预测虽容易计算但没有用处。

高斯密度由其均值和方差定义。使用高斯近似另一个密度相当于为这些参数选择合适的值。为了激励由拉普拉斯近似参数的选择，假设模型仅有一个参数 w 而不是两个参数，且我们知道 \hat{w} 是与最大后验一致的值。第一步是围绕最大值 \hat{w} 使用泰勒展开（见注解 4.3）近似 $\log g(w; \mathbf{X}, \mathbf{t}, \sigma^2)$ ：

$$\begin{aligned} \log g(w; \mathbf{X}, \mathbf{t}, \sigma^2) \approx & \log g(\hat{w}; \mathbf{X}, \mathbf{t}, \sigma^2) + \left. \frac{\partial \log g(w; \mathbf{X}, \mathbf{t}, \sigma^2)}{\partial w} \right|_{\hat{w}} \frac{(w - \hat{w})}{1!} \\ & + \left. \frac{\partial^2 \log g(w; \mathbf{X}, \mathbf{t}, \sigma^2)}{\partial w^2} \right|_{\hat{w}} \frac{(w - \hat{w})^2}{2!} + \dots \end{aligned}$$

注解 4.3 (泰勒展开)：泰勒展开是近似函数的一种方法。近似总是产生一些“大约”值——当远离那个值时，近似将趋于远离真实函数。有关 \hat{w} 的 $f(w)$ 泰勒级数定义为：

$$f(w) = \sum_{n=0}^{\infty} \frac{(w - \hat{w})^n}{n!} \left. \frac{\partial^n f(w)}{\partial w^n} \right|_{\hat{w}}$$

其中 $\left. \frac{\partial^n f(w)}{\partial w^n} \right|_{\hat{w}}$ 是 \hat{w} 处与 w 一致的 $f(w)$ 的 n 阶导数。当 $n=0$ 时，这个导数就是函数 $f(w)$ 。

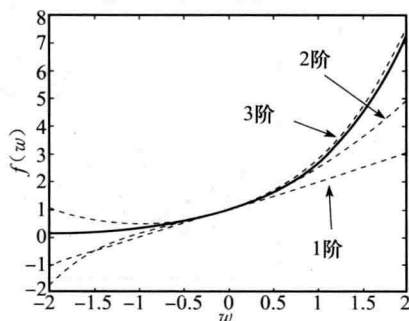
如果仅计算有限项，那么我们对函数有一个近似。1 阶近似仅包括项 $n=0, n=1$ — n 阶近似包括所有的阶次一直到 n 。例如，可以在 $\hat{w}=0$ 处近似 $f(w)=\exp(w)$ ：

$$\exp(w) = \exp(\hat{w}) + \frac{w}{1!} \exp(\hat{w}) + \frac{w^2}{2!} \exp(\hat{w}) + \dots$$

现在， $\exp(\hat{w})=1$ 所以，

$$\exp(w) = 1 + \frac{w}{1!} + \frac{w^2}{2!} + \frac{w^3}{3!} + \dots$$

当添加更多的项时，近似会越来越接近。这可以在右图中看出。



① 从技术上说，它实际上是一个鞍点近似，但在机器学习中已经称为拉普拉斯近似。在统计计算中，拉普拉斯近似完全是一个给其他东西的名字。

第二项是最大值点处的 1 阶导数（例如，倾斜度），因此必须为 0。删除该项，忽略 3 阶以及更高阶导数的项，剩下以下的表达式：

$$\log g(w; \mathbf{X}, t, \sigma^2) \approx \log g(\hat{w}; \mathbf{X}, t, \sigma^2) - \frac{v}{2}(w - \hat{w})^2 \quad (4-12)$$

其中 v 是 $w = \hat{w}$ 处 $\log g(w; \mathbf{X}, t, \sigma^2)$ 2 阶导数的负：

$$v = - \left. \frac{\partial^2 \log g(w; \mathbf{X}, t, \sigma^2)}{\partial w^2} \right|_{\hat{w}}$$

现在，高斯密度定义为：

$$\frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(w - \mu)^2\right\}$$

取对数后等于：

$$\log(K) - \frac{1}{2\sigma^2}(w - \mu)^2$$

其中， K 是归一化常数。这看起来与 $\mu = \hat{w}$ 和 $\sigma^2 = 1/v$ 的式 (4-12) 相似。这就是拉普拉斯近似——我们用高斯密度近似后验，这个高斯密度有后验模式 (\hat{w}) 的均值和反比于后验曲线（其 2 阶导数）的方差。

149

这个思想很容易扩展到多元密度。尤其是，拉普拉斯近似对真实后验 $p(w | \mathbf{X}, t, \sigma^2)$ 为：

$$p(w | \mathbf{X}, t, \sigma^2) \approx \mathcal{N}(\mu, \Sigma)$$

其中， μ 设置为 \hat{w} ， Σ 是 Hessian 矩阵逆的负：

$$\mu = \hat{w}, \Sigma^{-1} = - \left(\frac{\partial^2 \log g(w; \mathbf{X}, t)}{\partial w \partial w^T} \right) \Big|_{\hat{w}} \quad (4-13)$$

4.4.1 拉普拉斯近似实例：近似 γ 密度

在学习二值响应实例中的近似之前，有必要学习我们所知道的真实密度的一个例子（见练习 EX 4.1、EX 4.2 和 EX 4.3）（MARLAB 脚本：lapexample.m）。这允许我们看出近似有多好或多差。下面是随机变量 Y 的 γ 密度：

150

$$p(y | \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \exp\{-\beta y\} \quad (4-14)$$

我们将调查拉普拉斯近似对这个密度有多好。 γ 密度对其模式有一个分析表达式，这意味着我们不需要通过类似于第 3 章的优化过程。将这个模式 y 定义为：

$$\hat{y} = \frac{\alpha - 1}{\beta}$$

对 $p(y | \alpha, \beta)$ 的拉普拉斯近似采取高斯的形式：

$$p(y | \alpha, \beta) \approx \mathcal{N}(\mu, \sigma^2)$$

均值 μ 等于已经定义的 $p(y | \alpha, \beta)$ 的众数。为了找到近似高斯的方差 σ^2 ，我们需要找到 $\log p(y | \alpha, \beta)$ 对于 y 的 2 阶导数。它按以下计算：

$$\begin{aligned} \log p(y | \alpha, \beta) &= \alpha \log \beta - \log(\Gamma(\alpha)) + (\alpha - 1) \log y - \beta y \\ \frac{\partial \log p(y | \alpha, \beta)}{\partial y} &= \frac{\alpha - 1}{y} - \beta \\ \frac{\partial^2 \log p(y | \alpha, \beta)}{\partial y^2} &= -\frac{\alpha - 1}{y^2} \end{aligned}$$

σ^2 将等于 $y = \hat{y}$ 时这个数量的负逆。尤其是，

$$\sigma^2 = \frac{\hat{y}^2}{\alpha - 1} = \frac{\alpha - 1}{\beta^2}$$

在图 4-5 中, 可以看到 $p(y|\alpha, \beta)$ 和相应的拉普拉斯近似的两个例子。首先, $p(y|\alpha, \beta)$ 看起来更像是高斯函数, 而且得到近似值也很好。其次, $p(y|\alpha, \beta)$ 看起来非常不像高斯函数, 并且近似也不准确。在两种情况下, 当我们远离众数时, 近似就变得越差。这是因为近似值是基于众数点函数属性得到的。当回到二值响应模型时, 我们将再次看到这个特性。

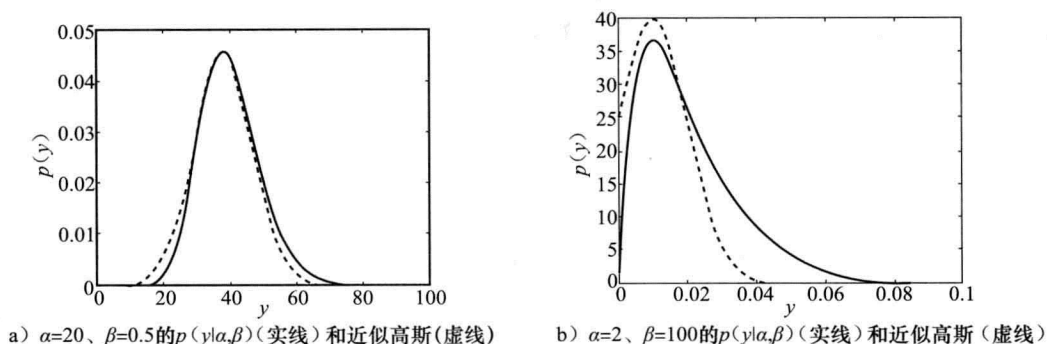


图 4-5 式 (4-14) γ 密度函数的拉普拉斯近似实例

4.4.2 二值响应模型的拉普拉斯近似

回到二值响应模型, 不得不计算两种众数, \hat{w} 和为牛顿-拉弗森过程的 Hessian 矩阵。因此我们已经做好了拉普拉斯近似后验 $p(w|X, t, \sigma^2)$ 的准备工作。在图 4-6a 中, 我们可以看到近似后验; 在图 4-6b 中, 在 $g(w; X, t)$ 顶端可以看到同样的近似值, 非归一化的后验。由于第 3 章的 γ 例子, 近似的形状能很好地围绕众数, 但是当我们远离众数时也远离了真实后验。这是预料期望的——拉普拉斯近似仅仅在众数匹配形状 (曲线)。我们也可以从近似后验取得 w 的值, 观察与其一致的决策边界。这样的决策边界在图 4-7a 中画出。在这些边界中出现了许多变化, 虽然它们都能很好地划分类别。

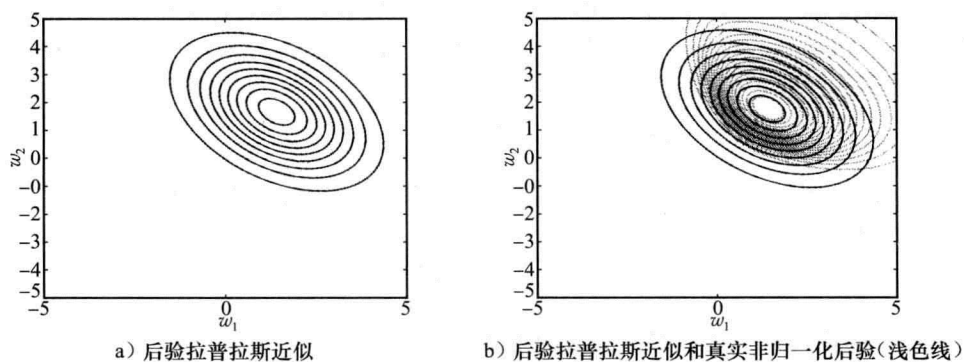


图 4-6 二值问题的拉普拉斯近似

最后一步是使用近似后验来计算预测。我们现在有一个 w 的密度而不是单一值, 从第 3 章中我们知道通过对这个密度进行平均计算期望值。尤其是, 我们应该计算与 w (我们记为 $\mathcal{N}(\mu, \Sigma)$) 近似后验相关的 $P(T_{\text{new}}=1 | x_{\text{new}}, w)$ 的期望值:

$$P(T_{\text{new}}=1 | x_{\text{new}}, X, t, \sigma^2) = E_{\mathcal{N}(\mu, \Sigma)} \{P(T_{\text{new}}=1 | x_{\text{new}}, w)\}$$

不幸的是, 我们不能计算期望中 w 的积分。这可能表明我们的近似选择不合理——我们仍然不能进行预测。然而, 可以简单地从 $\mathcal{N}(\mu, \Sigma)$ 中采样, 所以我们可以用以下公式近似期望 (见式 (2-23)):

$$P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, t, \sigma^2) = \frac{1}{N_s} \sum_{s=1}^{N_s} \frac{1}{1 + \exp(-\mathbf{w}_s^T \mathbf{x}_{\text{new}})} \quad (4-15)$$

其中, \mathbf{w}_s 是来自近似后验 N_s 样例中第 s 个。使用 $N_s = 1000$, 可以在图 4-7b 中看到 $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, t, \sigma^2)$ 的轮廓 (MATLAB 脚本: loglap.m)。和图 4-4b 进行比较。有很大的不同, 轮廓不再是直线。 \mathbf{w} 后验密度的均值对混淆决策边界有影响。在除了那些非常接近数据对象的所有区域中, 概率都接近 0.5。基于点估计的模型, 在图 4-4b 中, 可能被认为过一致——以 $x_1 = -3, x_2 = 5$ 为例子。按照点估计产生的预测 (见图 4-4b), 具有这些属性的对象可能有近似于 1 的概率, 尽管它离其他正方形对象非常远。把它与拉普拉斯近似后验一致的期望得到的概率近似为 0.6 的概率相比 (图 4-7b)。这个值看起来更合理。另一种理解出现在这个区域的不确定性, 见图 4-7a——在 $x_1 = -3, x_2 = 5$ 处可能的决策边界有很大变化。有些边界可能把对象划分为正方形, 有些则划分为圆形——它是一个正方形的概率, 我们已经看到给出的数据不是 1。

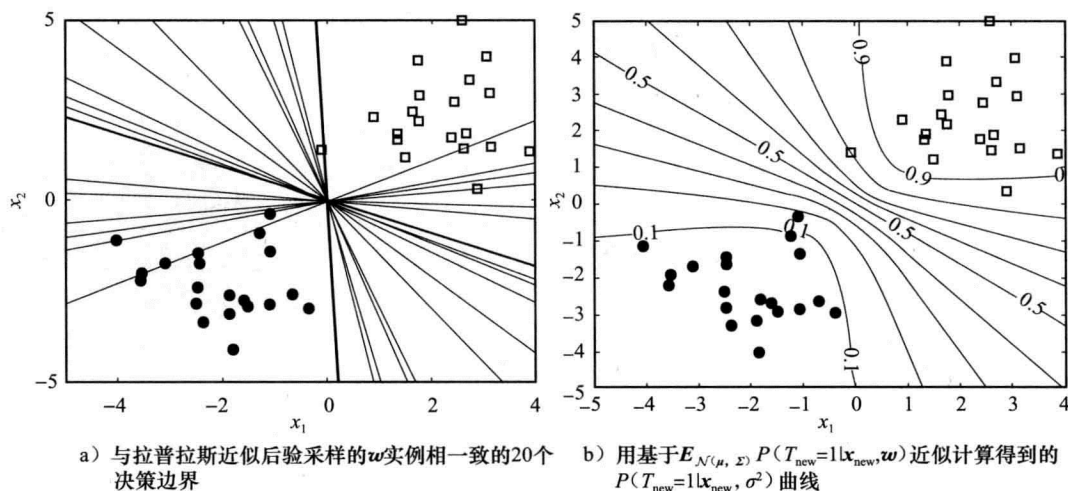


图 4-7 从拉普拉斯近似得到的决策边界和预测概率等概率率线

在本节中我们再次看到应该谨慎地使用点估计。这里所展示的拉普拉斯近似可以用来近似任何密度 (在实值随机变量上), 我们可以找到众数和计算 2 阶导数。这个方法假设后验可以被高斯合理地近似, 有些并不总是这样 (见图 4-5)。在二值响应模型中, 近似不允许我们精确地计算预测需要的期望。然而, 我们可以从高斯均值中采样, 得到基于样本近似的期望还是容易的。在 4.5 节中, 通过介绍一个技术来扩展这个思想, 使我们直接从 $p(\mathbf{w} | \mathbf{X}, t, \sigma^2)$ 中采样, 尽管我们不能计算归一化常数。产生这些样本的能力将允许我们使用基于样本近似的期望, 且不用近似后验。

4.5 抽样技术

4.4 节的拉普拉斯近似给我们提供了一个近似后验密度 $p(\mathbf{w} | \mathbf{X}, t, \sigma^2)$ 的方法。我们对后验密度感兴趣主要是因为预测时, 它考虑所有关于 \mathbf{w} 的不确定性因素。利用下面的期望:

$$P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, t, \sigma^2) = E_{p(\mathbf{w} | \mathbf{X}, t, \sigma^2)} \{P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})\}$$

求 \mathbf{w} 所有可能取值的平均值。在这个表达式中, 即使用近似值替代后验密度, 也不能用解析

方法计算出这个期望要的积分。幸运的是，可以很容易从高斯近似中抽样，使我们可以使用由式(4-15)计算出的基于近似的样本。在这个例子中，使用近似的好处是很容易生成样本。在本节段，我们着眼于一种技术，这种技术可以删除近似步骤，直接从后验密度中抽样。通过这种方式产生的真实后验的样本集能够直接代入式(4-15)，来计算想要的预测概率 $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \sigma^2)$ 。下面介绍一种流行的抽样技术——Metropolis-Hastings 算法。然而，在介绍它之前，先用一个不太抽象的例子说明抽样思想。

4.5.1 玩飞镖游戏

在投飞镖游戏中，玩家轮流地往飞镖盘上投飞镖，每人投三支飞镖，如图4-8所示。飞镖很锋利可以嵌入到飞镖盘中，每投一次飞镖，飞镖嵌入的位置决定玩家得到一个相应的分数，三只飞镖的得分加在一起，并从玩家当前的总分里扣除。每个玩家在游戏开始时都有相同的分数（通常是501分），先达到0分的玩家就是赢家。大部分的飞镖盘有20个段，如果飞镖落入这些段的白色区域，得分就等于段边缘标记的分数。如果飞镖落入阴影区，得分就是段边缘标记分数的两倍（外面浅色区域）或三倍（里面深色区域）。飞镖盘中心的同心圆叫做靶心或者牛眼，内部的圆形叫做内牛眼（50分），外边的圆形叫做外牛眼（25分）。这个游戏的规则稍微有些复杂，就是最后一镖必须投在双倍区并且使分数为0。例如，当前总分数是40分，玩家必须投到双倍区20分部分（标签“20”下面的浅色阴影区域），或者投一个20分（20分区域的白色段内），然后投到双倍区10分部分等。假设当前玩家有40分，同时只剩一只飞镖，换句话说，他需要投到20分双倍区才能赢，那么玩家能赢的概率是多少呢？

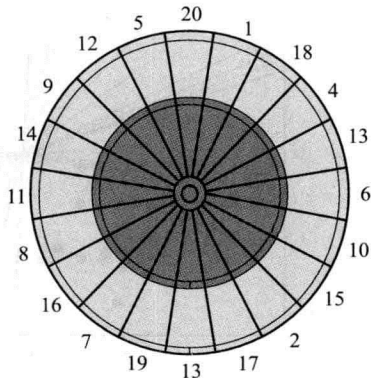


图 4-8 飞镖盘

假设有某个关于飞镖嵌入位置的概率密度函数，换句话说，玩家以双倍区20分为目标，飞镖的嵌入位置可以看成是某个变量的一个实例，用向量 \mathbf{y} 表示嵌入位置，则密度可以表示为 $p(\mathbf{y}|\Delta)$ ， Δ 在某些程度上依赖于玩家的目标，这种依赖程度取决于玩家的技术。如果一个专业玩家的目标是双倍区20分，那么嵌入位置可能会紧密集中在双倍区20分的周围。对于一个外行，目标对于最终的嵌入位置影响很小。 Δ 取决于玩家的技术以及技术的强度，因此 $p(\mathbf{y}|\Delta)$ 的值很难定义。

此时很容易让人放弃。但是，退一步讲，我们并不关心 $p(\mathbf{y}|\Delta)$ 的值，它只是玩家投到双倍区20分的概率。我们真的需要写出 $p(\mathbf{y}|\Delta)$ 的解析式来解决这个问题吗？在回答这个问题之前，先假设如果能写出 $p(\mathbf{y}|\Delta)$ 的值，就能算出玩家赢的概率。定义一个随机变量 $T=f(\mathbf{y})$ ，如果 \mathbf{y} 在双倍区20分内部， $f(\mathbf{y})$ 取1；否则， $f(\mathbf{y})$ 取0。 T 依赖于 \mathbf{y} ，因此依赖于 Δ 。因此，我们对 $P(T=1|\Delta)$ 概率感兴趣，这仅仅是一个期望值。尤其是，这看起来更像是上一节计算二值响应模型的期望值：

$$P(T=1|\Delta) = E_{p(\mathbf{y}|\Delta)}\{f(\mathbf{y})\} = \int f(\mathbf{y}) p(\mathbf{y}|\Delta) d\mathbf{y} \quad (4-16)$$

理论上，如果能计算出 $p(\mathbf{y}|\Delta)$ ，就能得出玩家赢的概率。然而，基于样本的近似我们还能计算大量的类似的问题。尤其是，假设从 $p(\mathbf{y}|\Delta)$ 中抽取 N_s 个样本， \mathbf{y}_s 定义为第 s 个样本，近似值可以表示为：

$$P(T=1|\Delta) \simeq \frac{1}{N_s} \sum_{s=1}^{N_s} f(\mathbf{y}_s)$$

因此, 想要计算 $P(T=1|\Delta)$ 的值, 只要能从 $p(y|\Delta)$ 中抽样, 就不必计算 $p(y|\Delta)$ 的值。幸运的是, 从 $p(y|\Delta)$ 中采样非常容易——已知玩家、飞镖、飞镖盘、玩家目标是双倍区 20 分。每只飞镖嵌入的位置就是从 $p(y|\Delta)$ 抽取的样本。如果记录每个 y_s , 就能计算出式 (4-16) 中基于样本的近似值。事实上, 这个例子就像是计算玩家投中双倍区 20 分的次数占总次数的比例。

可以将这个步骤关联到二值响应模型。首先, 在投飞镖这个例子中, 落入双倍区 20 分的概率 $P(T=1|\Delta)$ 类似于二值响应模型中的预测概率 $P(T_{\text{new}}=1|\mathbf{x}_{\text{new}}, \mathbf{X}, t, \sigma^2)$ 。在两个例子中, 要计算这个量, 就需要先给密度计算一个期望值: 投飞镖例子中的嵌入位置分布 $p(y|\Delta)$ 类似于二值响应模型中用参数表示的后验密度 $p(\mathbf{w}|\mathbf{X}, t, \sigma^2)$ 。在投飞镖例子中, 通过直接从后验密度抽样来近似期望; 在二值响应模型中, 通过抽样来近似后验密度。下面看看如何直接从 $p(\mathbf{w}|\mathbf{X}, t, \sigma^2)$ 抽样 (参见练习 EX 4.4)。

4.5.2 Metropolis-Hastings 算法

本节介绍 Metropolis-Hastings^① (MH) 算法。我们把这个算法当成一个秘诀来介绍, 并不过多介绍细节, 描述相关的步骤, 但对每步不做详细说明。在本节末尾提供了补充其他阅读资料。

记得我们想要从 $p(\mathbf{w}|\mathbf{X}, t, \sigma^2)$ 中抽样来近似下面的期望值:

$$\begin{aligned} P(T_{\text{new}}=1|\mathbf{x}_{\text{new}}, \mathbf{X}, t, \sigma^2) &= E_{p(\mathbf{w}|\mathbf{X}, t, \sigma^2)} \{P(T_{\text{new}}=1|\mathbf{x}_{\text{new}}, \mathbf{w})\} \\ &= \int P(T_{\text{new}}=1|\mathbf{x}_{\text{new}}, \mathbf{w}) p(\mathbf{w}|\mathbf{X}, t, \sigma^2) d\mathbf{w} \end{aligned}$$

其中,

$$P(T=1|\mathbf{x}_{\text{new}}, \mathbf{X}, t, \sigma^2) \simeq \frac{1}{N_s} \sum_{s=1}^{N_s} P(T_{\text{new}}=1|\mathbf{x}_{\text{new}}, \mathbf{w}_s)$$

156

Metropolis-Hastings 算法生成一个样本序列 $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{s-1}, \mathbf{w}_s, \dots, \mathbf{w}_{N_s}$ 。产生一个样本 \mathbf{w}_s 包括两步: 第一步, 提议一个新样本——作为 \mathbf{w}_s 的候选, 新样本通过移动前一个样本 \mathbf{w}_{s-1} 得到; 第二步, 测试提议样本, 观察其是否被接受, 如果提议样本被接受, 那么这个样本就确定为 \mathbf{w}_s ; 否则, 将新样本设置为 \mathbf{w}_{s-1} , 即 $\mathbf{w}_s = \mathbf{w}_{s-1}$ 。继续这个过程, 直到选到了足够的样本数。

现在, 如果我们提议样本是基于前一个样本的移动而得到的, 那么第一个样本 \mathbf{w}_1 该怎么得到呢? 实验证明起始点是不重要的, \mathbf{w}_1 可以是任何值。只要我们的抽样足够多, 抽样过程就保证结果收敛到一个让人感兴趣的分布。所以, 随机选一个 \mathbf{w}_1 (通过先验来抽样可能是明智的选择), 从 Metropolis-Hasting 算法开始, 直到算法收敛到一个正确的分布, 然后获取与需要一样多的样本。一句忠告: 理论上, 抽样器保证会收敛。实际上, 在开始获取样本之前用一种 (最好是多种) 有效的方法检测收敛性是很重要的。现在来看看提议步骤和接受步骤的详细过程。

提议一个新样本: 假设已经利用 MH 方法抽取了 $s-1$ 个样本。将要通过移动 \mathbf{w}_{s-1} 提议一个样本, 称为提出的样本 $\tilde{\mathbf{w}}_s$ 。(只有当 $\tilde{\mathbf{w}}_s$ 被接受了才可以称为 \mathbf{w}_s), 我们需要定义一个密度:

$$p(\tilde{\mathbf{w}}_s|\mathbf{w}_{s-1})$$

虽然我们从后验密度 $p(\mathbf{w}|\mathbf{X}, t, \sigma^2)$ 采样, 但是该密度不一定和它有任何联系, 我们想怎么定义就怎么定义。实际上, 该选择会影响 MH 算法收敛的时间。通常的选择是用高斯函数对当前样本 \mathbf{w}_{s-1} 进行处理:

$$p(\tilde{\mathbf{w}}_s|\mathbf{w}_{s-1}, \Sigma) = \mathcal{N}(\mathbf{w}_{s-1}, \Sigma)$$

① 以物理学家 Nicholas Metropolis 和统计学家 W. Keith Hastings 的名字命名。他们在物理学领域发明了一种叫做统计力学的处理问题的技术。

对值序列进行抽样就生成了众所周知的随机漫步。图 4-9 展示了两个随机漫步 (MATLAB 脚本: randwalk.m)。其中一个以 $w_1 = [0, 0]^T$ 为起点,

协方差 $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$; 同时另一个以 $w_1 = [2, 2]^T$

为起点, 协方差为 $\Sigma = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ 。后面的随

机漫步由于协方差矩阵对角元素很小, 所以每步移动很短的距离。正如曾经提到的, 利用高斯函数处理提议密度是常用的方法。一个原因就是可以轻松地进行抽样, 没必要选择一个难以抽样的提议分布, 这会导致问题变得复杂; 另一个原因是对称性, 从 w_{s-1} 减去 \tilde{w}_s 和把 \tilde{w}_s 加到 w_{s-1} 是一样的:

$$p(\tilde{w}_s | w_{s-1}, \Sigma) = p(w_{s-1} | \tilde{w}_s, \Sigma)$$

从下面的接受步骤中我们就可以看到这样做的优势。

接受或拒绝: 已知 \tilde{w}_s 是 w_s 的候选, 现在需要决定是接受 \tilde{w}_s 还是拒绝它。首先, 计算下面的比率:

$$r = \frac{p(\tilde{w}_s | X, t, \sigma^2)}{p(w_{s-1} | X, t, \sigma^2)} \frac{p(w_{s-1} | \tilde{w}_s, \Sigma)}{p(\tilde{w}_s | w_{s-1}, \Sigma)} \quad (4-17)$$

提议样本的后验密度除以前一个样本的后验密度再乘以提议密度的比率。上面提到的高斯函数提议分布的对称性允许将最后一项忽略, 因为它经常等于 1。第一项是后验密度用两种不同参数进行比较得到的比率。由于不能归一化表示它们, 所以不能准确地计算密度, 然而, 因为我们感兴趣的是比率, 所以归一化常量被抵消了。因此, 可以用前验比率乘以似然比率替代后验比率, 由此得到下面的表达式:

$$r = \frac{g(\tilde{w}_s | X, t, \sigma^2)}{g(w_{s-1} | X, t, \sigma^2)} = \frac{p(\tilde{w}_s | \sigma^2)}{p(w_{s-1} | \sigma^2)} \frac{p(t | \tilde{w}_s, X)}{p(t | w_{s-1}, X)}$$

密度函数永远为正, 因为比率永远为正。如果比率 r 大于或等于 1, 就接受样本 ($w_s = \tilde{w}_s$); 如果 r 小于 1, 接受样本的概率就是 r 。换句话说, 如果提出一个参数集合, 其对应的后验概率值比 w_{s-1} 大, 就接受; 否则, 就要视情况而定。算法如图 4-10 所示。我们非常详细地描述了接受/拒绝步骤。如果 $r < 1$, 接受的可能性为概率 r 。通过从 $0 \sim 1$ 之间的均匀分布抽取值 u 来实现, 因为是均匀分布的, 所以 $u \leq r$ 的概率是 r 。因此, 如果 $u \leq r$, 就接受提议; 否则, 就拒绝。可以用一个例子很好地解释整个过程。

图 4-11 显示了 Metropolis-Hastings 算法的操作过程, 抽取一个任意的密度 (用等概率线表示) (MATLAB 脚本: mhexample.m)。起始点 w_1 如图 4-11a 所示, 提议密度是 $\Sigma = I$ 的高斯函数。从起始点开始, 建立第一个提议 \tilde{w}_2 , 如图 4-11b

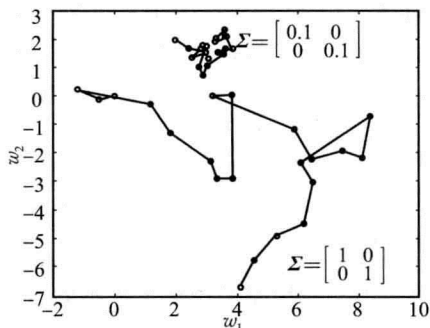


图 4-9 随机漫步的两个例子, 以当前位置为中心应用高斯函数得到下一个位置的分布, 两个随机漫步有不同的协方差矩阵, 图中用点标注出来

157

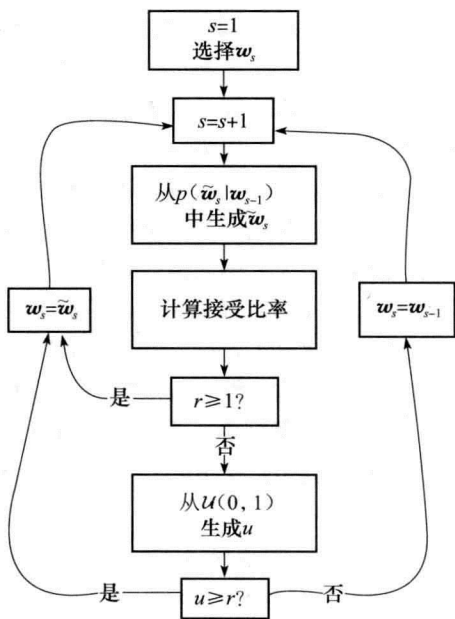


图 4-10 Metropolis-Hastings 算法

所示。提议导致了后验密度增长，因此接受 \tilde{w}_2 ，即 $w_2 = \tilde{w}_2$ 。这个接受过程如图 4-11b 中实线所示。下一个提议 \tilde{w}_3 引起后验密度的轻微下降，尽管如此， \tilde{w}_3 也被接受了（如果提议引起后验密度下降，提议仍然有被接受的可能），如图 4-11c 中新的实线所示。下一个提议 \tilde{w}_4 引起后验密度大幅下降，这样的提议很可能就不被接受了（比率远远小于 1），在这个例子中 \tilde{w}_4 就没被接受，如图 4-11c 虚线所示。因此， $w_4 \neq \tilde{w}_4$ ， $w_4 = w_3$ 。这个过程继续，如图 4-11d、e 所示，得到 10 个样本。按照这个过程，有三个提议被拒绝，它们的值被赋成前一个被接受样本的值。继续这个抽样过程，直到有 300 个样本被接受，如图 4-11f 所示。这些样本与密度等概率线相一致——看起来向密度中心集中，边缘部分很稀疏。

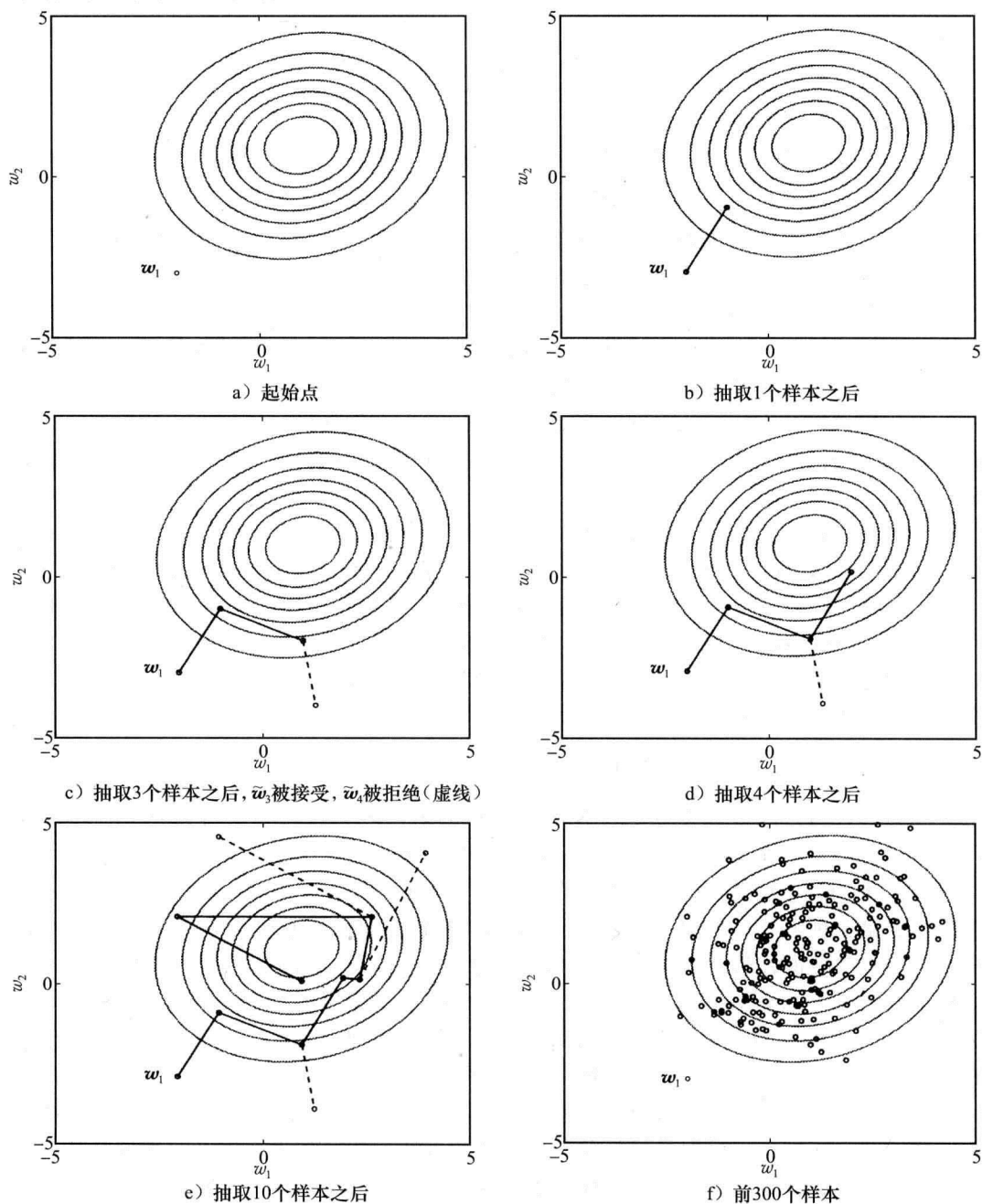


图 4-11 运行中的 Metropolis-Hastings 算法例子

这个例子中，我们抽样得到的密度正好是高斯函数。通过计算样本的均值和协方差，并看看它们与真实密度的均值和协方差是否一致，说明我们确实是从一个正确的密度中抽样的。真实的均值和协方差为：

$$\boldsymbol{\mu} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{S} = \begin{bmatrix} 3 & 0.4 \\ 0.4 & 3 \end{bmatrix}$$

通过 $N_s=10\,000$ 次抽样，计算出均值和协方差的基于样本的近似值：

$$\boldsymbol{\mu}' = \frac{1}{N_s} \sum_{s=1}^{N_s} \boldsymbol{w}_s, \boldsymbol{S}' = \frac{1}{N_s} \sum_{s=1}^{N_s} (\boldsymbol{w}_s - \boldsymbol{\mu}')(\boldsymbol{w}_s - \boldsymbol{\mu}')^T$$

得出的结果是：

$$\boldsymbol{\mu}' = \begin{bmatrix} 0.9770 \\ 1.0928 \end{bmatrix}, \boldsymbol{S}' = \begin{bmatrix} 3.0777 & 0.4405 \\ 0.4405 & 2.8983 \end{bmatrix}$$

这两个值都与真实值非常接近。

在将 MH 算法应用到二值响应模型之前，需要讨论两个相关的概念——**老化和收敛**。由于抽样过程可以从任何值开始（对 \boldsymbol{w}_1 没有限制），所以在启动抽样器的时候，没有必要规定启动位置（可能启动位置是后验密度非常低的区域）。因此前几个样本可能不具有代表性，应该删掉。从起始点到抽样器收敛的过程称为老化周期，这个阶段要持续多长时间是不能人为确定的。在上面描述的例子中，只有几个样本，在某些应用中经常有数百个或数千个例子。我们要控制收敛来克服这个问题，并不是要收敛到某个值，而是要收敛到一个特定的分布。换句话说，我们看到的样本是从正确的分布中得到的吗？

一个比较流行的方法是从不同的起始点同时启动多个抽样器，当所有的抽样器产生了具有相似属性（均值、方差等）的样本时，表明它们都收敛到同一个分布，这就是我们要抽样的分布。

现在回到二值响应模型。利用上面描述的 MH 方法从 $p(\boldsymbol{w} | \boldsymbol{X}, \boldsymbol{t}, \sigma^2)$ 中抽取 10 000 个样本（MATLAB 脚本：logmh.m），提议密度是 $\boldsymbol{\Sigma} = \gamma^2 \boldsymbol{I}$ 的高斯函数，其中 $\gamma^2 = 0.5$ 。在图 4-12a 中显示的是每 10 个样本画一条后验等概率线（把 10 000 个点都画出来会非常拥挤），样本和等概率线看起来合理清晰。如果喜欢的话，我们也可以使用样本生成具有两个特定参数的边缘后验密度，记得 3.4.1 节，为了从后验边缘化 w_2 ，需要对所有的 w_2 值进行积分（如果随机变量是离散的，就相加）：

$$p(w_1 | \boldsymbol{X}, \boldsymbol{t}, \sigma^2) = \int p(w_1, w_2 | \boldsymbol{X}, \boldsymbol{t}, \sigma^2) dw_2$$

这里 $p(w_1, w_2 | \boldsymbol{X}, \boldsymbol{t}, \sigma^2)$ 是 $p(\boldsymbol{w} | \boldsymbol{X}, \boldsymbol{t}, \sigma^2)$ 的另一种写法。为了得到基于样本的近似值，需要用到每一个样本 \boldsymbol{w}_s ，忽略 w_2 。换句话说，从每个样本里删除 w_2 的值，剩下的就是从 $p(w_1 | \boldsymbol{X}, \boldsymbol{t}, \sigma^2)$ 中抽样得到的集合。在图 4-12b~图 4-12d 中，用三种方式使这些样本可视化。第一种方式，如图 4-12b 所示，把可能值的阈值分成 20 段，计算落在每个段内的样本总数。黑色的柱形代表 w_1 的数目，灰色的柱形代表 w_2 的数目。如果想要把一定数目的样本放入某一段内，用总的样本数除以这个数目，得到的数字可以认为是 w_1 （或者 w_2 ）落入某一段的后验概率。第二种方式（见图 4-12c），把 10 000 个 w_1 样本全部描绘出来（描绘 w_2 的点和描绘 w_1 的点基本一致），这种图让我们相信抽样器可以快速收敛。如果没有快速收敛，可能会有总体增长或下降的趋势。图 4-12d 展示了两个适合于样本的连续密度函数。这三种方式展示了机器学习任务各种可能的解决方法，如果样本看起来像是从一个高斯函数得到的，那么我们可以给两个样本集合规定高斯密度（见练习 EX 2.8）。在这个例子中，我们使用了核密度估计，这是一种更常用的技术。在 Matlab 中用概率密度分布函数

(ksdensity) 来计算。这里就不再详细地讨论了——因为有许多使样本可视化的方法，可以把样本转化为连续密度函数（近似的）。

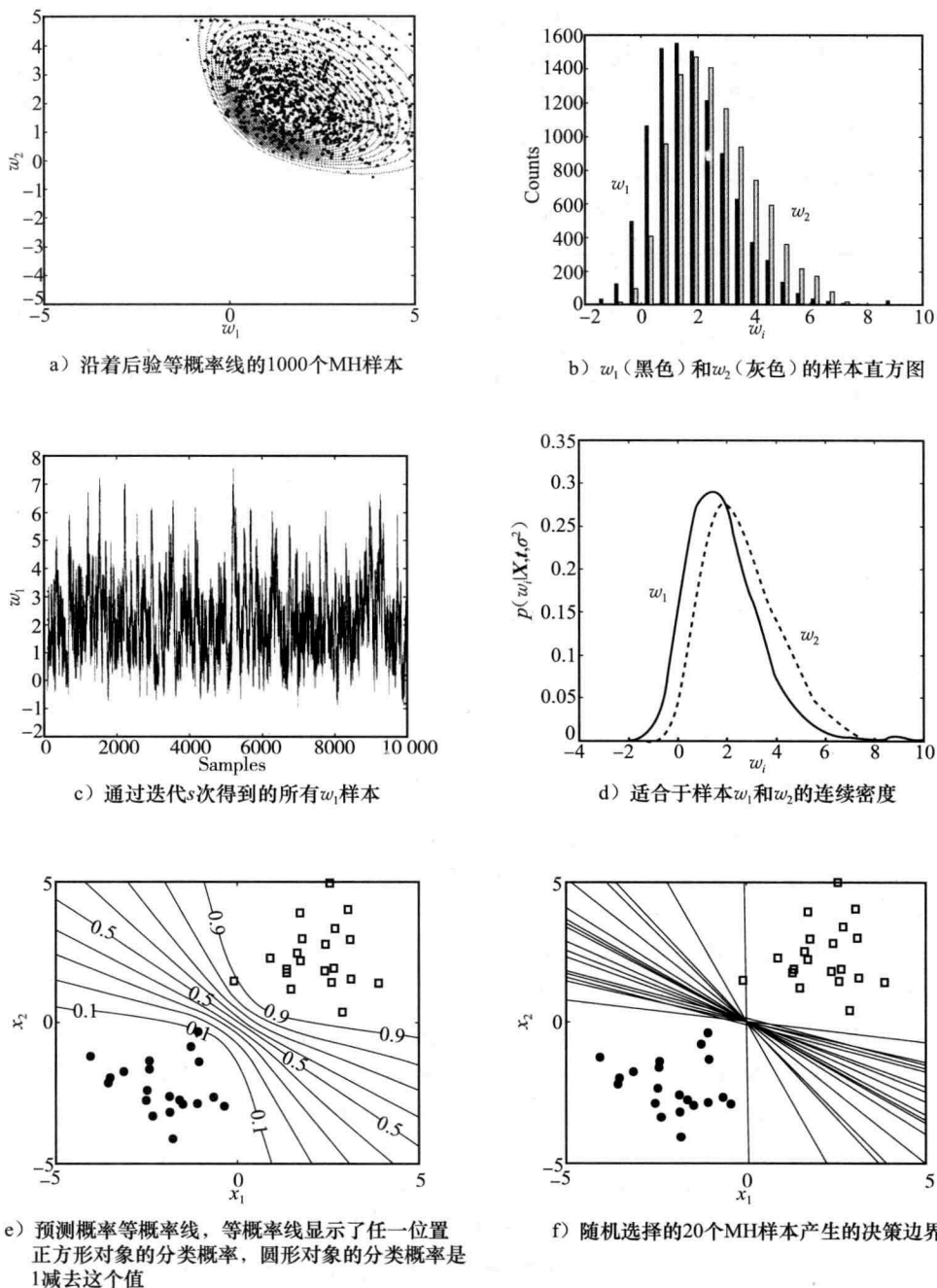


图 4-12 将 MH 抽样算法应用于二值响应模型的结果

最后，把注意力转回概率预测， $P(T_{\text{new}}=1 | \mathbf{x}_{\text{new}}, \mathbf{X}, t, \sigma^2)$ 。当采用拉普拉斯近似的时候，通过从近似后验中抽取样本 w_1, \dots, w_{N_s} ，然后用下面公式计算，可以得出近似值：

$$P(T_{\text{new}}=1 | \mathbf{x}_{\text{new}}, \mathbf{X}, t, \sigma^2) = \frac{1}{N_s} \sum_{s=1}^{N_s} \frac{1}{1 + \exp(-\mathbf{w}_s^T \mathbf{x}_{\text{new}})}$$

现在已经有了真实后验的样本集，可以用同样的方法计算。图 4-12e 显示了用这些真实

后验样本计算得到的预测概率等概率线。任何位置上的类似于正方形的对象，等概率线都给出了这些对象的分类概率。等概率线的形状更像是图 4-7b 中的形状，这并不令人惊讶，因为如图 4-6b 所示，拉普拉斯近似看起来与真实后验非常相似。唯一值得注意的是，图 4-12e 中围绕着数据区域建立的等概率线有一点不紧凑，这表明概率下降得很慢，比从一个正方形到另一个正方形下降得更慢。MH 抽样器从真实后验中取样，所以图 4-12e 中的等概率线比图 4-16a 中拉普拉斯近似的等概率线更接近真实值，这个比较只是想指出拉普拉斯逼近在预测的时候性能有多好。图 4-12f 显示了对应着随机选取的 20 个 MH 样本（见图 4-7a）的决策边界（见练习 EX 4.6、EX 4.7 和 EX 4.8）。

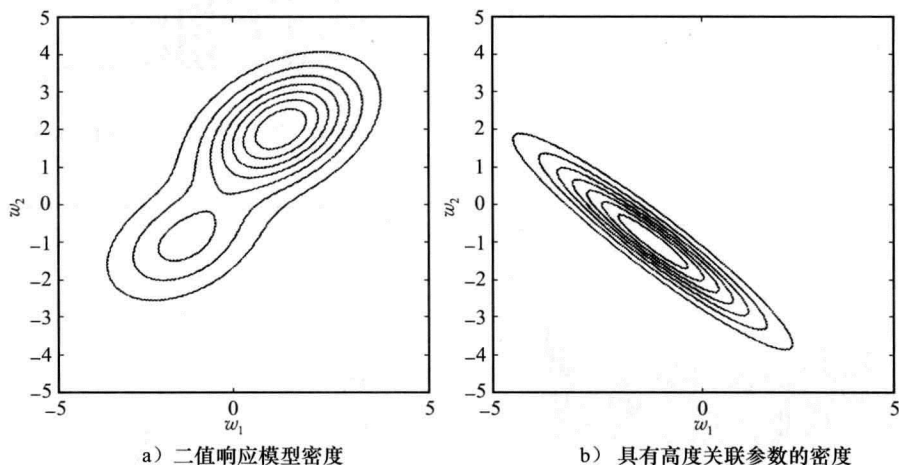


图 4-13 通过 MH 抽样得到的两个微妙的密度

4.5.3 抽样的艺术

Metropolis-Hasting 算法应用于二值响应模型看起来工作得很好，但并不总是这样——抽样方法（如 MH）使用起来非常微妙。难点在于我们要从中抽样的密度模型经常是未知的。考虑图 4-13a 所示的密度，有两个众数，一个是 $\mathbf{w} = [-1, -1]^T$ ，另一个是 $\mathbf{w} = [2, 2]^T$ 。MH 算法喜欢朝着众数移动，移动导致后验密度增加，因此样本容易被接受。想象 \mathbf{w}_s 在众数 $\mathbf{w} = [2, 2]^T$ 附近， \mathbf{w}_s 从这里移动到众数 $\mathbf{w} = [-1, -1]^T$ 需要在行方向上大幅下降。尽管有这种可能，但是可能性微乎其微。这就是类似于 MH 这类算法在理论上和实际上出现偏差的地方。理论上，从一个众数顶端的某一点移动到另一个众数顶端的某一点（可能性不大不代表不会发生）。实际上，我们可能会变得很老了。当发现其中一个众数的时候我们很开心，但是却没意识到另一个众数的存在。

第二个问题用图 4-13b 中的密度来说明，图中仅仅是一个众数的密度，但是两个变量 w_1 和 w_2 互相依赖或者说是高度关联的。如果 w_1 是已知的，那么就有可能把 w_2 限制在一个很小的范围内，像 $p(\tilde{\mathbf{w}}_s | \mathbf{w}_{s-1})$ 这样的密度很难选择提议分布。选取图 4-13b 中密度的任何位置，想象对一个对角协方差矩阵（与我们用到的所有例子一样）提议一个基于高斯密度的移动，这个矩阵在描绘的时候有圆形的等概率线。图 4-13b 显示密度并不是圆形的，因为提议密度模型与我们试图从中抽样的密度模型有很大不同，所以许多样本会被拒绝：大部分提议抽样的移动会加剧概率下降倾斜度。

还不仅仅是这些问题。例如，我们怎么才能知道什么时候抽取了足够的样本呢？怎么才能知道要丢弃多少个最初阶段的样本呢？幸运的是，有许多方法可以克服这些问题：更高级的算法、选择提议密度的方法，表明收敛的数量估计等。其他阅读材料中提供了更多的细节。

4.6 小结

当我们不能用解析方法计算分布的时候, 希望用贝叶斯方法解决问题, 这就是本章的动机。我们已经列举了三个通用技术的例子。首先, 找到后验的最高点 (MAP 估计), 这是个单一值, 单一值并不是准确的贝叶斯方法, 但是它结合了先验知识, 因此它被认为是对最大似然解的改进。第二个方法是用另一个密度来近似后验, 我们选择拉普拉斯近似, 它使用高斯函数近似后验。在许多应用中, 这个密度能够用解析方法计算想要的期望值。在二值响应模型应用中, 期望值难以用解析方法处理, 但是从高斯函数抽样是很简单的, 所以利用基于抽样的近似方法。第三个方法, 利用 Metropolis-Hastings 算法从用来计算期望的真实后验中产生样本, 这导致了额外的计算代价, 但是 (至少在理论上) 我们得到了反映真实后验的预测结果。

4.7 练习

EX 4.1 包含 N 个观察对象 \mathbf{x}_n 的数据集 (每个 \mathbf{x}_n 是 D 维的), 真实值是 t_n , 线性回归模型如下定义:

$$p(t_n | \mathbf{x}_n, \mathbf{w}) = \mathcal{N}(\mathbf{w}^T \mathbf{x}_n, 1)$$

进行标准 IID 假设, 假设有 D 维参数 \mathbf{w} 的高斯先验, 说明拉普拉斯近似等于真实后验。

EX 4.2 第3章计算了硬币正面朝上概率 r 的后验密度, 使用了 β 先验和二项式似然, 具有参数 α 和 β 的 β 先验为:

$$p(r | \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1} (1-r)^{\beta-1}$$

假设 N 次抛硬币有 y 次正面朝上的二项式似然为

$$p(y | r, N) = \binom{N}{y} r^y (1-r)^{N-y}$$

计算后验的拉普拉斯近似。(注意: 你应该能够获取 MAP 值的一个闭型解 \hat{r} , 通过设置对数后验为 0, 微分并等于 0)。

EX 4.3 描绘真实的 β 后验和练习 EX 4.2 中具有各种 α 、 β 、 y 和 N 值的拉普拉斯逼近。

EX 4.4 给定一个圆形区域的表达式, $A = \pi r^2$, 利用唯一的非均匀分布的随机变量, 设计一个计算 π 值的抽样方法。

EX 4.5 重新整理逻辑函数:

$$P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \hat{\mathbf{w}}) = \frac{1}{1 + \exp(-\hat{\mathbf{w}}^T \mathbf{x}_{\text{new}})}$$

证明 $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \hat{\mathbf{w}}) = 0.5$ 时, $\hat{\mathbf{w}}^T \mathbf{x}_{\text{new}} = 0$ 。

EX 4.6 假设观察到具有 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 的 N 个向量, 以及相关的整数计数 t_1, \dots, t_N 。可以应用泊松似然:

$$p(t_n | \mathbf{x}_n, \mathbf{w}) = \frac{V_n^{t_n} \exp\{-V_n\}}{t_n!}$$

其中, $V_n = \exp(\mathbf{w}^T \mathbf{x}_n)$ 。

假设 \mathbf{w} 的高斯先验, 要获取倾斜度和 Hessian 矩阵, 需要用到牛顿-拉弗森过程来查找参数 \mathbf{w} 的 MAP 解。

EX 4.7 导出练习 EX 4.6 中模型的拉普拉斯近似值。

EX 4.8 实现练习 EX 4.6 中模型的一个 Metropolis-Hasting 抽样机制, 比较后验和练习 EX 4.7 中的拉普拉斯近似值。

其他阅读材料

- [1] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50:5–43, 2003.

从机器学习的角度，是一本从特殊点到 MCMC 技术的介绍性教程。

- [2] Siddhartha Chib. Understanding the Metropolis-Hasting algorithm. *The American Statistician*, 49(4):327-335, 1995.

一本出色的 Metropolis-Hasting 算法教程，深入了解整个算法家族的良好开始。

- [3] Arnaud Doucet, Nando de Freitas, and Neil Godron, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2010.

本书中没有包括序列蒙特卡罗技术，但它们对于在复杂模型中进行贝叶斯推理越来越流行，特别是带有时空组件的模型，如目标跟踪。

- [4] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, second edition, 2004.

这是对实际贝叶斯推理的一个很好资源。特别是，它提供了其他抽样技术的介绍，以及如果 Metropolis-Hastings 和其他抽样算法收敛时的检测过程。

- [5] W.R. Gilks, S. Richardson, and D. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, 2005.

一本提供了有趣实际抽样实例的册子。

- [6] Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123-214, 2011.

作者近期的一篇论文，描述了复杂分布形式下抽样的一种精致的 Metropolis 算法。

167

- [7] Jun Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2008.

- [8] Carl Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

在这两本书中，我们简要介绍了无参高斯过程替代有参模型。书中对于分类和回归高斯过程的应用进行了全面的介绍。

- [9] Simon Rogers, Richard Scheltema, Mark Girolami, and Rainer Breitling. Probabilistic assignment of formulas to mass peaks in metabolomics experiments. *Bioinformatics*, 25(4):512-518, 2009.

作者近期的一篇论文，描述了将贝叶斯抽样方法（Gibbs 抽样）应用到质谱实验中检测代谢产物的问题。

- [10] Michael Tipping and Alex Smola. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211-244, 2001.

一个基于线性模型回归和分类的例子，在分类实例中使用了 Laplace 近似方法。

- [11] Christopher Williams and David Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342-1351, 1998.

有关使用高斯过程分类的第一篇论文之一，真实近似贝叶斯推理的一个很好介绍。

168

分 类

前几章主要介绍了机器学习方法的主要概念。对于一个特定模式，我们已经了解到如何选择参数以及如何对观测数据做出预测。主要有 3 种方法：找出使误差最小的参数、找出使似然函数最大的变量和将参数变为变量的方法。在本章以及以后的章节中，我们将会再次看到其中的一些方法。因为我们要解决机器学习领域主要的算法家族：分类、聚类和预测。

本章主要解决分类问题。机器学习领域有很多分类算法，它们在每天的基础上增长。我们有选择地引进 4 个算法。这 4 个算法通常被认为是分类技术的基础知识，通过对它们的学习，可以使读者有能力解决一些基本的分类问题，也可以更进一步探讨其他领域知识。

这 4 个算法可以分为两种类型——那些输出为概率的和输出非概率的。这两种类型都有自己的优势，选择永远依赖于数据集。

5.1 一般问题

通常情况下，数据集有 N 个训练对象， $\mathbf{x}_1, \dots, \mathbf{x}_n$ 。每个对象都是一个 D 维向量。对于每个对象，我们还提供了一个标签 t_n 描述对象 n 属于哪个类别。 t_n 通常会取整数值。例如，如果数据分为两类， $t_n = \{0, 1\}$ 或 $t_n = \{-1, 1\}$ 。通常情况下，如果有 C 个类别，则 $t_n = \{1, 2, \dots, C\}$ ，我们的目标是对于给定的对象 \mathbf{x}_{new} ，预测它的类别 t_{new} 。

有必要将这一章的内容与第 1 和第 3 章进行对比。在前面两个章节中，我们提供了一组对象 $\mathbf{x}_1, \dots, \mathbf{x}_n$ 并给它们赋值为实数。对于许多例子，对象是奥运会的年数和男子 100 米获胜的时间。我们的目标是对于未来的奥运会比赛预测获胜时间。分类设定都非常相似——分类的响应变量是一个表明某个类的整数而不是实数。事实上，在第 4 章中我们已经看到了分类的一个例子。二值响应模型是一个众所周知的二值分类算法，称为逻辑回归。

分类算法已成功地应用在许多领域。两个极具挑战性的例子是自动疾病诊断和文本分类。在自动疾病诊断中，主要根据医疗观察预测病人是否健康。在文本分类中，主要为特定的用户根据主题与相关性对文本进行分类。这两个例子说明分类技术应用领域的多样化。不同的领域有它们自己的相关问题。例如，在第一个例子中，如何处理错误的不平衡代价？在第二个例子中，如何处理复杂的数据对象，如文本？这些问题将在后面的章节中解决。

169

5.2 概率分类器

概率和非概率分类器的不同在于输出的类型不同。概率分类器产生的是对于一个新对象，其属于某个特定类别的概率。用矩阵和向量 (\mathbf{X}, \mathbf{t}) 的形式表达训练数据，对于类别 c 来说，它的概率是

$$P(T_{\text{new}} = c | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) \quad (5-1)$$

作为概率，它必须满足以下两个条件：

$$\begin{aligned} 0 &\leq P(T_{\text{new}} = c | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) \leq 1 \\ \sum_{c=1}^C P(T_{\text{new}} = c | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) &= 1 \end{aligned}$$

以概率作为输出看起来可能没有必要。我们刚刚说我们的任务是预测类别 T_{new} 。如果我们只对分配问题感兴趣,那么我们可能会选择一个非概率分类。然而,在许多应用中,概率分类是非常有用的,因为它在输出中提供了一个置信度水平。例如,考虑两个类别的疾病诊断系统,健康用 0 表示、患病用 1 表示。提供概率 $P(T_{\text{new}}=1 | \mathbf{x}_{\text{new}}, \mathbf{X}, t)$ 比简单地说明 $t_{\text{new}}=1$ 更实用。 $P(T_{\text{new}}=1 | \mathbf{x}_{\text{new}}, \mathbf{X}, t)=0.6$ 和 $P(T_{\text{new}}=1 | \mathbf{x}_{\text{new}}, \mathbf{X}, t)=0.9$ 都表明 \mathbf{x}_{new} 应该归类为患病。但前者的确定性更小。在做出一个决定之前也许要做更多的测试。

5.2.1 贝叶斯分类器

贝叶斯分类器是已知的第一个概率分类器,由它所依据的方程式而得名。对于 C 个类别的给定训练样本点,我们首先计算 C 个类别的先验概率(式(5-1))。这些概率可以作为决策过程的基础(如分配 \mathbf{x}_{new} 为概率最高的类别),也可以用来计算期望值。

从贝叶斯规则中(参见本书 2.2.7 节和第 3、4 章),我们可以得到一个基于先验概率的表达式:

$$P(T_{\text{new}}=c | \mathbf{x}_{\text{new}}, \mathbf{X}, t) = \frac{p(\mathbf{x}_{\text{new}} | T_{\text{new}}=c, \mathbf{X}, t) P(T_{\text{new}}=c | \mathbf{X}, t)}{p(\mathbf{x}_{\text{new}} | \mathbf{X}, t)}$$

边缘似然函数 $p(\mathbf{x}_{\text{new}} | \mathbf{X}, t)$ 可以展开为 C 个类别的总和,得到贝叶斯分类公式:

$$P(T_{\text{new}}=c | \mathbf{x}_{\text{new}}, \mathbf{X}, t) = \frac{p(\mathbf{x}_{\text{new}} | T_{\text{new}}=c, \mathbf{X}, t) P(T_{\text{new}}=c | \mathbf{X}, t)}{\sum_{c'=1}^C p(\mathbf{x}_{\text{new}} | T_{\text{new}}=c', \mathbf{X}, t) P(T_{\text{new}}=c' | \mathbf{X}, t)} \quad (5-2)$$

接下来的任务是计算 $p(\mathbf{x}_{\text{new}} | T_{\text{new}}=c, \mathbf{X}, t)$ 和 $P(T_{\text{new}}=c | \mathbf{X}, t)$, \mathbf{x}_{new} 属于第 c 个类别的似然函数和第 c 个类别的先验概率。在后面的章节中,我们将一一计算。

5.2.1.1 似然函数——类条件分布

式(5-2)中的似然函数 $p(\mathbf{x}_{\text{new}} | T_{\text{new}}=c, \mathbf{X}, t)$ 是第 C 个类别的分布函数(由 $T_{\text{new}}=c$ 来决定),用来计算 \mathbf{x}_{new} 。虽然目前还没有原因为什么会这样,但是通常对每个类都使用相同类型的分布。对于分布函数的选择,应该尽量依赖被建模数据的类型和我们已知的任何额外的知识。一旦我们对第 c 个类别选定了分布函数,剩下的问题就是选择参数。例如,如果我们选择了高斯分布函数,就需要决定均值和方差(协方差)。类别为 c 的分布函数的参数将只用来训练 c 类的数据。这个阶段可以被看做是一种机器学习的问题,我们将在 5.2.1.3 节进一步讨论。

5.2.1.2 类分布的先验知识

式(5-2)的第二个变量是 $P(T_{\text{new}}=c | \mathbf{X}, t)$ 。这是在现有训练数据 \mathbf{X}, t 上,目标属于 c 类的概率。它使我们在看到数据之前,能够指定 \mathbf{x}_{new} 的先验类别。这使我们考虑到类别大小不均匀的情况。例如,也许有一个类别 c 几乎为空。在看到数据之前,我们可能偏向于它不属于类别 c (使 $P(T_{\text{new}}=c | \mathbf{X}, t)$ 非常低)。这样只有当 \mathbf{x}_{new} 的似然概率特别高时,才将它分为 c 类。另外, c 类可能很小,但它可能是误分类某些稀有实例的关键,我们总想要检测它。在这种情况下,将 $P(T_{\text{new}}=c | \mathbf{X}, t)$ 赋予一个很高的值。这将导致更多潜在 \mathbf{x}_{new} 向量被划为 c 类。当然其中一些将是不正确的(实际上,它们属于另一个类),但我们不会错过许多真正属于 c 类。我们在先验概率的基础上做出决策时也可以解决这些问题。我们将在 5.4 节详细讨论这些问题。

无论我们的动机是什么,在选择 $p(T_{\text{new}}=c | \mathbf{X}, t)$ 时的唯一技术限制是它们大于零和 $\sum_c P(T_{\text{new}}=c | \mathbf{X}, t)=1$ 。两个常用的选择是:

- 1). 均匀先验: $P(T_{\text{new}}=c | \mathbf{X}, t) = \frac{1}{C}$ 。

2) 基于类别大小的先验: $P(T_{\text{new}}=c|\mathbf{X}, t) = \frac{N_c}{N}$, 其中 N 是训练集的对象数目、 N_c 是属于 c 类的对象数量。

注意, 虽然我们写出 \mathbf{X} 和 t 条件下的先验, 但是它不一定依赖于 \mathbf{X} 和 t 。上面两个例子在定义先验时都没有使用 \mathbf{X} , 只有第二个例子用到了 t (通过 N_c)。

5.2.1.3 高斯类条件分布的分类举例

图 5-1 中显示的数据产生于 3 个类别。每个训练对象为一个 2 维的属性向量 $\mathbf{x}_n = [x_{n1}, x_{n2}]^T$ 和相关的标签 $t_n = \{1, 2, 3\}$ 。类 1 用黑色圆圈标记, 类 2 用白色菱形标记, 类 3 用灰色方块标记。考虑到属性为实数, 我们将使用高斯类条件分布:

$$p(\mathbf{x}_n | t_n = c, \mathbf{X}, t) = \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \quad (5-3)$$

其中 $\boldsymbol{\mu}_c$ 和 $\boldsymbol{\Sigma}_c$ 需要根据类 c 的训练点给定。我们将这些点标注为 \mathbf{X}^c 。这就是一种机器学习的任务——我们有一些数据 (\mathbf{X}^c), 并希望从中推断出有关模型的参数。在这个例子中, 我们要找出使观测值 \mathbf{X}^c 的似然函数最大化的 $\boldsymbol{\mu}_c$ 和 $\boldsymbol{\Sigma}_c$ 。我们也可以使用贝叶斯方法作为替代。例如, 定义参数的先验密度 $p(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ 。根据贝叶斯规则可以计算出后验概率:

$$p(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c | \mathbf{X}^c) = \frac{p(\mathbf{X}^c | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) p(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{p(\mathbf{X}^c)}$$

然后通过下面的期望值计算 \mathbf{x}_{new} 的似然函数:

$$p(\mathbf{x}_{\text{new}} | T_{\text{new}} = c, \mathbf{X}, t) = E_{p(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c | \mathbf{X}^c)} \{p(\mathbf{x}_{\text{new}} | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)\}$$

172

假设将先验概率 $p(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ 的选择与高斯似然函数相结合, 后验概率和期望都可以从分析中得到。当数据很少时, 使用贝叶斯分析可以达到最好的效果, 因此 $\boldsymbol{\mu}_c$ 和 $\boldsymbol{\Sigma}_c$ 的期望值是不确定的。请阅读练习 EX 5.1 和 EX 5.2。

具有 N 个数据点的集合的高斯分布的均值和协方差的最大似然估计可以从对每个参数的自然对数求微分得到, 设置为零就可以解决 (如在第 2 章求解线性模型一样)。省略细节 (参见练习 EX 5.3), 最大似然估计是:

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{n=1}^{N_c} \mathbf{x}_n \quad (5-4)$$

$$\boldsymbol{\Sigma}_c = \frac{1}{N_c} \sum_{n=1}^{N_c} (\mathbf{x}_n - \boldsymbol{\mu}_c)(\mathbf{x}_n - \boldsymbol{\mu}_c)^T \quad (5-5)$$

其中求和公式只针对 c 类数据的实例。三个类条件分布如图 5-2 所示 (MATLAB 脚本: plotcc.m)。

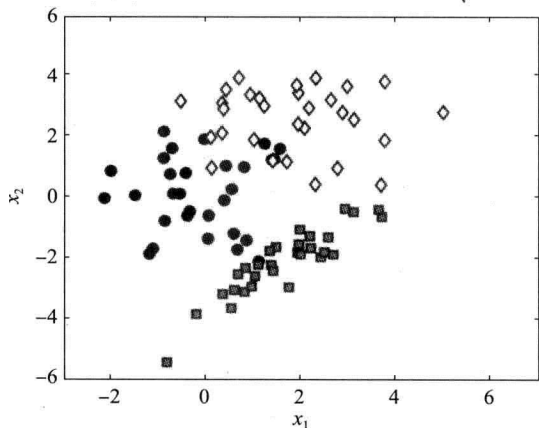


图 5-1 三类分类数据集

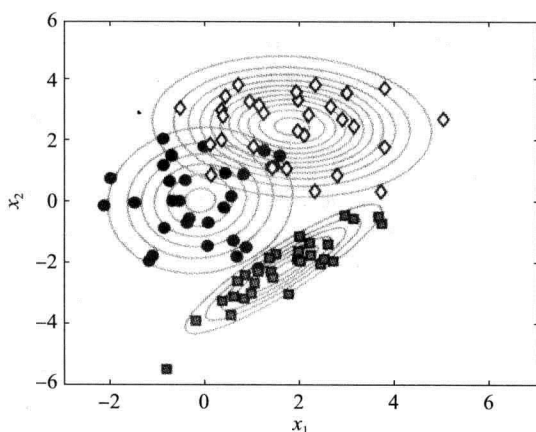


图 5-2 应用式 (5-4) 和式 (5-5) 对密度服从三个类条件分布的数据集分类

剩下的任务是确定先验概率 $P(T_{\text{new}}=c|\mathbf{X}, t)$ 。正如前面提到的, 通常选择 $P(T_{\text{new}}=c|\mathbf{X}, t)=\frac{N_c}{N}$, 即 c 类中训练点的比例。在我们的例子中, 每类的 $N_c=30$, 因此 $P(T_{\text{new}}=c|\mathbf{X}, t)=\frac{1}{3}$ 。

5.2.1.4 预测

有了类条件分布和先验概率后, 就可以做预测了。作为一个例子, 我们将计算 $\mathbf{x}_{\text{new}}=[2, 0]^T$ 的后验类条件概率。对将要计算的各个变量求和, 它们的 \mathbf{x}_{new} 值在表 5-1 中。最后一列给式 (5-2) 的分子。为了转化值到概率, 我们必须用三个值的总和 ($0.0046+0.0020+0.0001=0.0067$) 除以每个值。结果概率为:

$$P(T_{\text{new}}=1|\mathbf{x}_{\text{new}}, \mathbf{X}, t)=0.6890$$

$$P(T_{\text{new}}=2|\mathbf{x}_{\text{new}}, \mathbf{X}, t)=0.3024$$

$$P(T_{\text{new}}=3|\mathbf{x}_{\text{new}}, \mathbf{X}, t)=0.0087$$

表 5-1 对于 $\mathbf{x}_{\text{new}}=[2, 0]^T$ 时高斯条件贝叶斯分布例子的似然和先验

c	$p(\mathbf{x}_{\text{new}} T_{\text{new}}=c, \boldsymbol{\mu}, \boldsymbol{\Sigma}_c)$	$P(T_{\text{new}}=c \mathbf{X}, t)$	$p(\mathbf{x}_{\text{new}} T_{\text{new}}=c, \boldsymbol{\mu}, \boldsymbol{\Sigma}_c)P(T_{\text{new}}=c \mathbf{X}, t)$
1	0.0138	$\frac{1}{3}$	0.0046
2	0.0061	$\frac{1}{3}$	0.0020
3	0.0002	$\frac{1}{3}$	0.0001

从中我们可以看出, \mathbf{x}_{new} 属于 1 类 (黑圈) 的概率是属于 2 类 (白色菱形) 两倍, 而且它不可能属于 3 类 (灰色正方形)。

通过许多 \mathbf{x}_{new} 的值评估分类概率, 就能画出分类概率的等概率线图。这能从图 5-3 中看出 (MATLAB 脚本: bayesclass.m)。对于每一类, 模型分配一个很高的概率给该类训练点构成的空间区域。然而, 有些奇怪的效果。如图 5-3a 所示: 在区域的中左边, 大多数 1 类 (黑圈) 数据分布的地方, 有大于 0.9 的高概率。然而, 在图的底部右边, 没有 1 类数据 (或来自任何类的数据) 的地方也有高概率。类似地, 图 5-3b 中的 2 类等概率线, 在没有数据属于 2 类的图形的右中部有高概率。这些效果能通过观察图 5-2 中, 与 1 类和 2 类进行比较的 3 类条件分布的陡峭度来解释。它的密度比 1 类和 2 类衰退得很快, 到了 3 类的右边。1 类和 2 类的密度函数都相对较高。这是一个不好的特性, 它似乎对图右边属于 1 类或 2 类的高概率标签点不敏感。它将变得更好, 如第 4 章中的二值响应模型, 当我们从数据附近移动时, 概率将变得更不确定。

5.2.1.5 朴素贝叶斯假设

在以前的例子中, 对于条件分布我们使用 2 维高斯。这些分布能够获得每级属性变量之间的依赖关系。例如, 我们看到 3 类条件分布能获得对于试验点存在于 x_1 和 x_2 之间的强依赖关系。拟合 2 维高斯包括选择 5 个参数值: 两个对于 $\boldsymbol{\mu}$ 、3 个对于 $\boldsymbol{\Sigma}$ ($\boldsymbol{\Sigma}$ 是对称的, 所以不在对角线上的元素相等)。在每类中有 30 个试验点, 这是极其可行的。当维度增加时, 问题开始出现。一般地, D 维高斯要求 $D + D + \frac{D(D-1)}{2}$ 个参数 (D 对应于均值, $D + \frac{D(D-1)}{2}$ 对应于协方差矩阵)。对于 10 维, 30 个点不能可靠地拟合 65 个参数。

部分克服这个问题的一种方法 (缺乏数据仅仅能靠增加数据来解决) 是进行朴素贝叶斯假设: D 维类条件分布能分解为 D 个单变量分布。换句话说, 在特殊类上加条件, 维度 (如 x_1 和 x_2) 是独立的。单变量高斯要求两个参数: μ 和 σ^2 。因此, D 维要求 $2D$ 个参

数——当维度为 10 时这种方法比原来所需要的高斯参数能够减少 45 个。参数减少的代价是模型灵活性的降低。在高斯例子中，它意味着限制了类条件分布和轴对称的形状——在维度互相依赖的情况下不能再使用此模型。从图 5-4 中可以看得很清楚，当进行朴素贝叶斯假设时，我们看到了类条件分布的密度等概率线：

$$p(\mathbf{x}_n | t_n = k, \mathbf{X}, t) = \prod_{d=1}^2 p(x_{nd} | t_n = k, \mathbf{X}, t)$$

与图 5-2 比较，清楚地看到 3 类模型不能准确地反映数据特性。图 5-5 描述了 3 个类别的分类概率等概率线。有趣的是，尽管我们知道类条件分布对于 3 类不是特别适合，但分类等概率线仍然是合理的（尽管当我们移动数据时缺乏不确定性）。

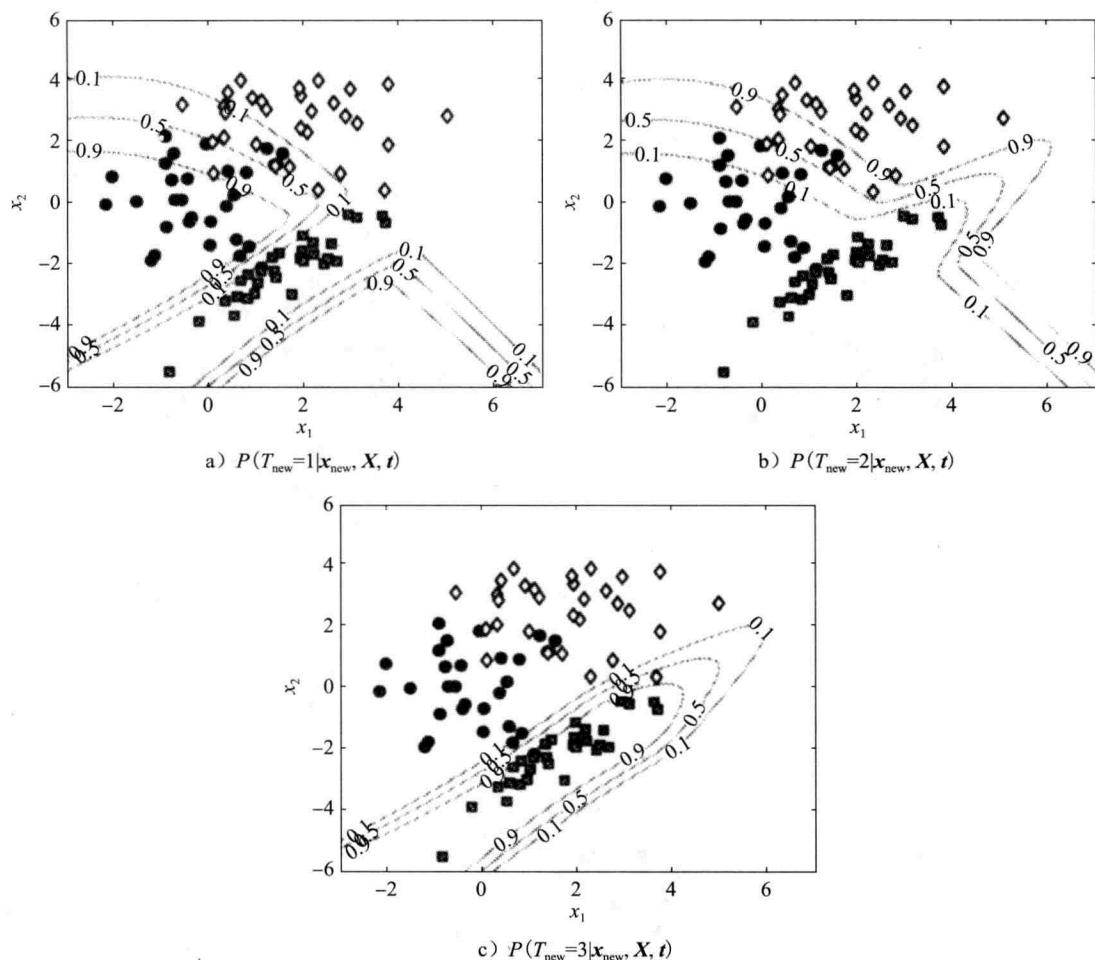


图 5-3 应用贝叶斯分类器对高斯条件分布数据分类的结果的等概率线图

5.2.1.6 例子——对文本进行分类

机器学习广泛地用于自动文本分类。在这个领域中从数据学习很有意义——它不是直接人工构建一系列的能够用于对文本进行分类的规则或模型，而是使用分类器同时对大量数据进行训练。

20 个新闻组数据集是常用的基准数据集，用它来评估新算法。它包括大约 20 000 个文件，每个文件中存储 20 个新闻组。考虑将这 20 个新闻组作为不同的类，构建一个自动将一个文件分配到这 20 类之一的分类系统。这些组涵盖了一系列不同的主题，其中包括运动、

计算和宗教。

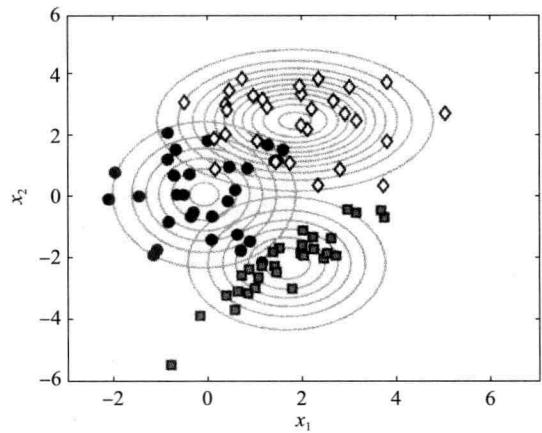


图 5-4 具有朴素贝叶斯假设的高斯类条件分布的密度等概率线

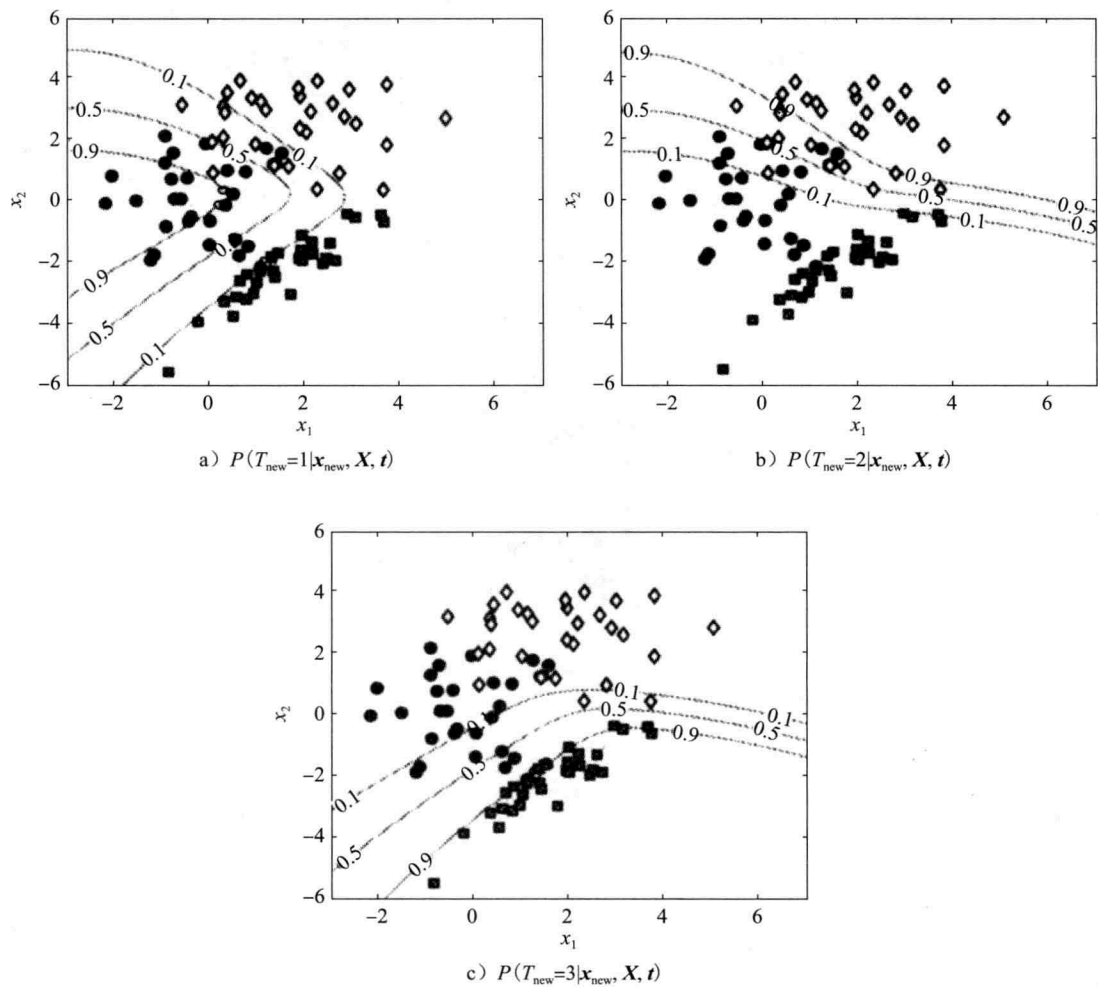


图 5-5 具有高斯类条件分布和朴素贝叶斯假设的贝叶斯分类器分类概率的等概率线图

算法用数值数据工作，因此我们需要一种方法将文本编码为数值矢量。这个领域中最常

用的方法是词袋模型。假如在所有文档（词汇表）中独特词的总数是 M ，每个文件用 M 维矢量表示。第 n 个文件的矢量 \mathbf{x}_n ，由每个单词出现的次数组成。 x_{nm} 是单词 m 在文件 n 中出现的次数。

因为词汇表可能很大，所以我们将做朴素贝叶斯假设。因此，类条件分布能够通过将词汇表中的词分解成下面的乘积：

177

$$p(\mathbf{x}_n | T_n = c, \dots) = \prod_{m=1}^M p(x_{nm} | T_n = c, \dots)$$

这意味着表示每个类条件分布所需要的参数个数与词的数量大致相同（取决于分布函数的选择）。词之间增加任何形式的依赖将导致我们要适应的参数数量明显扩大。例如，假如我们检查成对的依赖关系，那么需要 M^2 个参数的顺序。假设一个典型的词汇表包括大约 50 000 个单词，这已经是一个重大挑战。

词袋模型还假设单词的顺序不重要。例如， \mathbf{x}_n 在下面的两个句子里是一样的，尽管第二个句子没有意义：

1) The quick brown fox jumps over the lazy dog.

2) Dog quick lazy the jumps fox brown the over.

这个假设不太严格：假如分类器分类一个包括很多“baseball”的文件，很可能该文件是关于运动而忽略单词的特殊顺序。注意，词袋模型忽略顺序但是没有暗示独立。我们仍然能定义类条件分布，允许在 \mathbf{x}_n 的元素之间存在依赖。

对于类条件分布，我们将用多项式（在 2.3.3 节中介绍）。向量 \mathbf{x}_n 的多项式分布定义为：

$$P(\mathbf{x}_n | \mathbf{q}) = \left[\frac{s_n!}{\prod_{m=1}^M x_{nm}!} \right] \prod_{m=1}^M q_m^{x_{nm}} \quad (5-6)$$

这里 $s_n = \sum_{m=1}^M x_{nm}$ 和 $\mathbf{q} = [q_1, \dots, q_M]^T$ 是参数，每一个都是概率（ $\sum_m q_m = 1$ ）。注意，多项式分布通过 m 的乘积自动满足朴素贝叶斯假设。

每类都有一个多项式（因此一个 \mathbf{q} ）。因此，我们需要基于训练对象集 \mathbf{x}_n （对应于类 c ）确定 \mathbf{q}_c 的值（第 c 类别的概率向量）。我们能用最大似然估计实现它（见练习 EX 5.4），产生：

$$q_{cm} = \frac{\sum_{n=1}^{N_c} x_{nm}}{\sum_{m'=1}^M \sum_{n=1}^{N_c} x_{nm'}}$$

其中，仅在第 c 类中对 n 个数据求和。定义先验分布 $p(T_{\text{new}} = c | \mathbf{X}, t) = \frac{1}{C}$ ，使用式 (5-2) 进行预测。然而，在进行预测前有个问题需要说明。

5.2.1.7 平滑性

一个单词（如 m ）从不出现在一个类别（如 c 类）的文件中是合理的——不会有很多宗教的新闻组中提及“棒球”。这将导致 $q_{cm} = 0$ 。回看式 (5-6)——假如任何或更多的 $q_{cm} = 0$ 而且

177

$x_{nm} \neq 0$ ，那么乘积 $\prod_{m=1}^M q_{cm}^{x_{nm}}$ 将等于零。换句话说，我们尝试计算新文件 \mathbf{x}_{new} 的分类概率，碰巧包括单词 m ，那么似然概率（ $p(\mathbf{x}_{\text{new}} | T_{\text{new}} = c, \mathbf{q}_c)$ ）等于零，因此 $P(T_{\text{new}} = c | \mathbf{x}_{\text{new}}, \mathbf{X}, t) = 0$ 。如果文件包含没有出现在任何训练记录中的单词，则它将不属于任何类。这是对训练数据过拟

合的另外一个例子，我们可以通过给 q 增加一个先验密度来克服这个问题，使所有的概率都大于零。一旦我们定义了这个先验值，我们可以用 MAP 评估设置 q 值（见第 4 章）而不是根据最大似然评估。我们也能预测 q 的后验密度（见练习 EX 5.5 和 EX 5.6）。

对于概率向量的合适先验密度是狄利克雷（Dirichlet）密度，定义如下：

$$p(q_c | \alpha) = \frac{\Gamma(\sum_{m=1}^M \alpha_m)}{\prod_{m=1}^M \Gamma(\alpha_m)} \prod_{m=1}^M q_{cm}^{\alpha_m - 1} \quad (5-7)$$

我们将通过假设 $\alpha_m = \alpha$ 进一步简化该公式——这个参数说明狄利克雷密度对每个单词都一样。MAP 估计值可以通过对多项式似然函数的乘积（或乘积的对数）求最大化来获得（或乘积的日志）。这里省略详情（见练习 EX 5.7），对于 q_{cm} 的 MAP 估计是：

$$q_{cm} = \frac{\alpha - 1 + \sum_{n=1}^{N_c} x_{nm}}{M(\alpha - 1) + \sum_{m'=1}^M \sum_{n=1}^{N_c} x_{nm'}} \quad (5-8)$$

而且，求和也是仅对 c 类训练对象。对 $\alpha > 1$ ， $q_{cm} > 0$ 和零问题不再是一个问题。这种技术经常作为平滑的手段——如果我们持续增加 α 值，那么每个单词的概率 q_{cm} 将越来越接近 $\frac{1}{M}$ 。这也可以当做正则化的例子（见 1.6 节）。

新闻组数据被分为训练和测试部分，各为 11 000 和 7000 个记录。设置 $\alpha=2$ ，用式（5-8）确定 q_c ，设置先验分类概率为 $1/20$ （20 类的均匀先验），我们可以用式（5-2）计算分类概率，式（5-6）给出了 $p(x_{\text{new}} | t_{\text{new}}=c, \mathbf{X}, t)$ ， x_{new} 代替了 x_n 。

对于约 7000 个 x_{new} 向量中的每个向量来说，有 20 个概率的集合。评估分类器如何工作的最简单方法是将每个 x_{new} 分配到最高概率的一类，并将它们与已知的标签进行比较。如果我们这样做，我们可以发现分类器正确的概率为 78%——考虑到我们只是用了最简单的模型，而且还没有使用方法进行优化，这个结果已经很不错了。

图 5-6 提供了对大约 7000 个测试点的分类概率的图形表示（MATLAB 脚本：newspred.m）。每行对应一个单一测试点，行按真实类排序。每列相应一个预测类。例如，第 10 列的值给出了属于第 10 类的测试点的概率。呈现的块状结构告诉我们算法是合理的——概率和它们的真实值一样高。从图中可以明显地看出数据是否分类错误。例如，属于 19 类的大量测试点（倒数第二块）被错误地分为 17 类。这两类来自新闻组 talk.politics.guns 和 talk.politics.misc。难怪这儿有些疑惑——很多流行词将被这两类共享。另一个迷惑是对于在 20 类和 16 类之间的数据点，谁的真类是 20。这两类来自 talk.religion.misc 和 soc.religion.christian，它们也明显相关。分类算法错误类型的分析能提高分类的性能。在

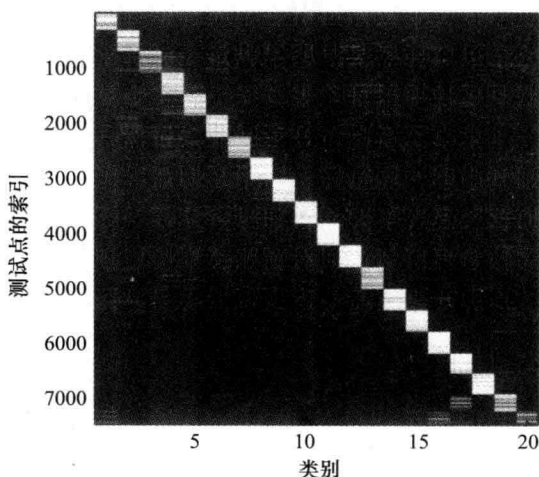


图 5-6 关于贝叶斯分类器对 20 个新闻组数据预测概率的图形表示。每行对应一个测试点，测试点按真实类顺序排列，颜色越白，概率越高

这个例子中, 16 类和 20 类是否应该融合到一个大类是敏感的。如果不融合, 我们就应该从这两类中尝试获得更多的数据(文件)。

有许多方法对分类算法产生的结果进行分析。我们将在之后的章节(5.4 节)更加详细地讨论这些方法。同时, 我们将研究第二类概率分类。

5.2.2 逻辑回归

尽管我们称它为二值响应模型, 但第 4 章将专门介绍称为逻辑回归的二值分类。在第 4 章我们没有从分类的观点真正地讨论它, 而是作为模型分析贝叶斯推理不可行。这种方法的详细介绍在第 4 章, 这里将不再重复。然而, 有几个问题值得讨论——我们称为“压缩函数”的动机和模型的泛化。

179

5.2.2.1 动机

在第 4 章, 我们提出了逻辑似然概率

$$P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_{\text{new}})}$$

我们想使用熟悉的线性模型 ($\mathbf{w}^T \mathbf{x}$), 但是需要对其进行转换使得输出为一个概率 ($0 \leq P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w}) \leq 1$)。

同时, 将它作为动机没有任何错误, 但是将逻辑似然概率转换为对数差异比作为最终结果通常更为正式。这是 $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})$ 和 $P(T_{\text{new}} = 0 | \mathbf{x}_{\text{new}}, \mathbf{w})$ 之间的比率对数:

$$\log\left(\frac{P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})}{P(T_{\text{new}} = 0 | \mathbf{x}_{\text{new}}, \mathbf{w})}\right)$$

对这个值没有限制——它能取任何实际值。如果 $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w}) \ll P(T_{\text{new}} = 0 | \mathbf{x}_{\text{new}}, \mathbf{w})$, 那么这个对数比率将取一个大的负数; 如果 $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w}) \gg P(T_{\text{new}} = 0 | \mathbf{x}_{\text{new}}, \mathbf{w})$, 那么它将取很大的正值。因此, 当采用我们熟悉的线性模型建模时, 采用这种数量关系是非常有用的:

$$\log\left(\frac{P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})}{P(T_{\text{new}} = 0 | \mathbf{x}_{\text{new}}, \mathbf{w})}\right) = \mathbf{w}^T \mathbf{x}_{\text{new}} \quad (5-9)$$

重新排列, 并注意

$$P(T_{\text{new}} = 0 | \mathbf{x}_{\text{new}}, \mathbf{w}) = 1 - P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})$$

对于 $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})$, 我们能得到一个表达式:

$$\log\left(\frac{P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})}{P(T_{\text{new}} = 0 | \mathbf{x}_{\text{new}}, \mathbf{w})}\right) = \mathbf{w}^T \mathbf{x}_{\text{new}}$$

$$\frac{P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})}{P(T_{\text{new}} = 0 | \mathbf{x}_{\text{new}}, \mathbf{w})} = \exp(\mathbf{w}^T \mathbf{x}_{\text{new}})$$

$$\frac{P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})}{1 - P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})} = \exp(\mathbf{w}^T \mathbf{x}_{\text{new}})$$

$$P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})(1 + \exp(\mathbf{w}^T \mathbf{x}_{\text{new}})) = \exp(\mathbf{w}^T \mathbf{x}_{\text{new}})$$

$$P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w}) = \frac{\exp(\mathbf{w}^T \mathbf{x}_{\text{new}})}{1 + \exp(\mathbf{w}^T \mathbf{x}_{\text{new}})}$$

$$P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_{\text{new}})}$$

通过使用 $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})$ 的逻辑似然, 我们用线性模型对对数差异比建立模型。在统计中, 像这种社区方法叫做广义线性模型——通过线性模型进行转换来构建需要的变量。

180

5.2.2.2 非线性决策函数

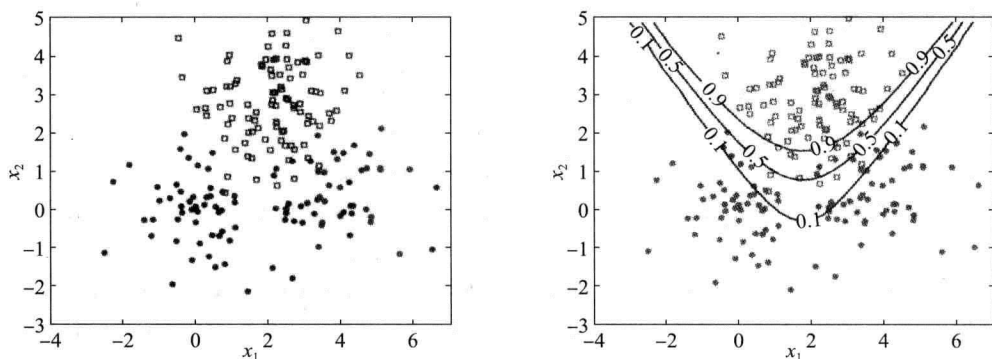
在第 4 章中, 对于单个 \mathbf{w} 值的决策边界都是直线。通过拉普斯近似和 Metropolis-Hastings

算法获得的概率等概率线被刻画成很多直线求平均的结果。通过扩展 \mathbf{x}_n 将 x_n^2 包含进来, 我们可以使用第1章使用的非线性回归方法, 得到类似该方法的逻辑回归的非线性决策边界。例如, 图5-7a中的数据显示了可能要求非线性决策边界的二值分类数据库。

用 x_1 和 x_2 来表示单一属性 ($\mathbf{x}=[x_1, x_2]^T$), 我们可以用下面模型计算对数差异比:

$$\log\left(\frac{P(T_{\text{new}}=1|\mathbf{x}_{\text{new}}, \mathbf{w})}{P(T_{\text{new}}=0|\mathbf{x}_{\text{new}}, \mathbf{w})}\right) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 \quad (5-10)$$

为了说明这可能产生非线性决策边界, 我们找到了 MAP 评估参数的 $\hat{\mathbf{w}}$, 假设高斯先验 $p(\mathbf{w}|\sigma^2) = \mathcal{N}(\mathbf{0}, \sigma^2 I)$ (见4.3节)。当然, 假如需要的话, 我们还可以将这个模型如第4章所述的多种贝叶斯方法来处理。



a) 用非线性决策边界能够更好地对数据库建模

b) 拥有2阶术语模型的概率(见式(5-10))。这些基于参数的点评估($\hat{\mathbf{w}}$)用牛顿-拉弗森优化获得

图5-7 式(5-10)描述的逻辑回归模型的二值数据和分类等概率值线

将 $\hat{\mathbf{w}}$ 引入逻辑似然允许我们计算分类概率:

$$P(T_{\text{new}}=1|\mathbf{x}_{\text{new}}, \hat{\mathbf{w}}) = \frac{1}{1 + \exp(-\hat{\mathbf{w}}^T \mathbf{x}_{\text{new}})}$$

评估 \mathbf{x}_{new} 值的网格使我们能够在属性空间画出属于类别1的概率等概率线。图5-7b中显示了非线性决策边界(为使轮廓清晰, 训练点是灰色的)(MATLAB 脚本: nonlinlogreg.m)。这是逻辑回归最有吸引力的特性, 但是必须记住: 在第1章中, 当使用的线性模型越来越复杂时会出现过拟合和非普适的问题, 这些问题在分类领域中同样棘手。记住这是对于单值 \mathbf{w} : 在第4章中, 在拉普拉斯和 Metropolis-Hastings 下获得非线性决策边界, 但是仅仅是通过对许多不同的直线求平均。

5.2.2.3 非参数模型——高斯过程

在本书中, 我们把模型限制为 $\mathbf{w}^T \mathbf{x}$ 的形式。这个模型有一组线性 \mathbf{w} 参数。对 \mathbf{x} 的特殊增大(如加平方项), 这个函数属于一个特殊的函数家族。例如, 假如其中含有平方项, 则它是平方家族的一员。家族的选择限制了函数的灵活性——假如我们选择 $w_0 + w_1 x$, 那么我们只能构建直线模型。如果选 $w_0 + w_1 x + w_2 x^2 + w_3 x^3$, 那么我们只能构建立方(3阶)多项式模型。这样的模型都是参数模型, 因为它们属于特殊的参数家族, 这里的特殊函数由一组 \mathbf{w} 参数值确定。

值得简单一提的是, 另一种非常灵活的选择——非参数模型。与定义为某些参数的函数(如 $f(\mathbf{x}, \mathbf{w})$) 不同, 非参数模型以一种通用的方式定义。例如, 常用的非参数模型是高斯过程(GP)。在参数模型中, 首先定义关于先验 \mathbf{w} 的分布, 不同的 \mathbf{w} 分布代表了结果的不同先验分布。使用高斯过程, 我们直接得到函数输出值的先验分布。注意非参数并不意味着高

斯过程不需要任何参数，而是对函数不假设参数形式。

高斯过程由两个函数描述——均值函数 $\mu(\mathbf{x})$ ，描述作为属性 \mathbf{x} 函数的平均函数值（ x 可以是标量或矢量）；协方差函数 $c(\mathbf{x}_n, \mathbf{x}_m)$ ，定义在 \mathbf{x}_n 处函数的输出与 \mathbf{x}_m 处输出的相似度。实际上，均值函数通常假设为 0。

对于任何含有 N 个数据点的有限集合，高斯过程本质上成为 N 维高斯分布问题，其均值为 $\boldsymbol{\mu} = [\mu(\mathbf{x}_1), \dots, \mu(\mathbf{x}_N)]^T$ ，协方差矩阵为：

$$\mathbf{C} = \begin{bmatrix} c(\mathbf{x}_1, \mathbf{x}_1) & c(\mathbf{x}_1, \mathbf{x}_2) & \cdots & c(\mathbf{x}_1, \mathbf{x}_N) \\ c(\mathbf{x}_2, \mathbf{x}_1) & c(\mathbf{x}_2, \mathbf{x}_2) & \cdots & c(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ c(\mathbf{x}_N, \mathbf{x}_1) & c(\mathbf{x}_N, \mathbf{x}_2) & \cdots & c(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

对于 N 个数据对象的每一个的函数输出值，依据该高斯模型对向量进行采样。

由于高斯过程的灵活性，它在机器学习中已变得越来越普及——高斯过程没有限定为某一特殊的参数族。在任何情况下它们都可以替代参数模型。例如，我们能用高斯过程对奥运数据建模或者在逻辑回归算法中替代 $\mathbf{w}^T \mathbf{x}_n$ 。这里省略详细的讨论，但是鼓励读者在机器学习领域深入研究高斯过程的使用。

182

5.3 非概率分类器

现在我们将注意力转向非概率分类器。不同于概率分类器 $P(T_{\text{new}} = c | \mathbf{x}_{\text{new}}, \mathbf{x}, \mathbf{t})$ 提供一个类别归属的可能性，它们的输出是一个对象对一个类别的指定： $t_{\text{new}} = c$ 。我们将关注两个不同的算法—— K 近邻算法（K-Nearest Neighbours, KNN）和支持向量机（Support Vector Machine, SVM）。由于它们杰出的实验表现，它们在机器学习中都深受欢迎。支持向量机还提供给我们关于核方法领域的介绍。

5.3.1 K 近邻算法

首先我们来看看 K 近邻算法，简单的思想和良好的实验表现使它深受欢迎。它可以同时处理二值和多类别数据，并不对决策边界的参数形式进行假设。KNN 没有训练过程，可以通过对新对象 \mathbf{x}_{new} 进行分类的简单过程来对其进行最好的描述。

考虑一个常见的场景——我们有 N 个训练对象，每个对象都可以描述为属性 \mathbf{x}_n 和标记 t_n 的集合。为了利用 KNN 对 \mathbf{x}_{new} 进行分类，我们首先找到距离 \mathbf{x}_{new} 最近的 K 个训练点。然后将 t_{new} 设置为这些邻居节点个数最多的分类。如图 5-8 所示。训练数据由属于两种类别（灰色圆圈和白色方块）之一的数据点组成。两个测试点标记为黑色菱形，虚线所画的圆圈围住了 $K=3$ 个最近的邻居。测试点 A 的邻居包含方块类的 2 个和圆圈类中的一个，所以将它分类为方块类。测试点 B 的所有邻居都属于圆圈类，因此也将 B 分为此类。

KNN 算法的一个缺点是，当两个或更多的类别拥有相同个数投票时的判定问题。例如，如果在图 5-8 中 $K=8$ ，那么对于每个类别我们都拥有 4 个邻居而分不出多少。一种解决办法就是从这些类别集合里随即指定一个类别。这种方法并不是总是合理的，因为它意味着对于相同的 \mathbf{x}_{new} 如果测试多次，

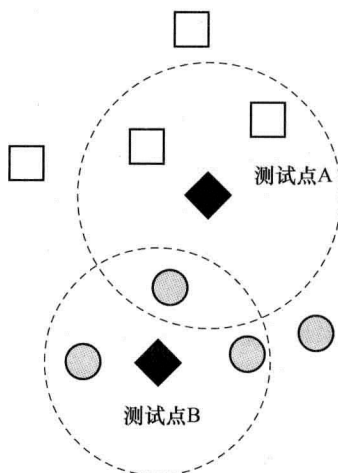


图 5-8 KNN ($K=3$) 效果演示图。圆圈和方块代表训练点，菱形代表测试点。测试点 A 被认定为方块类，B 被认定为圆圈类

可能得到不同的分类结果。对于二值分类器，一个最简单的方法就是总选取奇数个邻居节点。更一般地，可以根据邻居节点的距离来给投票加上权值，即距离越近点的投票影响越大。

在图 5-8 中，我们根据几何距离来决定哪些点属于测试点的邻居。然而，我们可以选择任意我们喜欢的距离测量标准。因此 KNN 算法是很灵活的——它可以用于任意可以定义两个对象之间距离的数据格式。KNN 成功运用的其他数据格式的例子包括字符串、图形和图像。

5.3.1.1 选择 K

一旦我们有了一些数据并选择了一种可行的定义距离的方法，剩下唯一的工作就是 K 的选取。如果 K 太小，分类器就很容易被噪声干扰。如图 5-9a (MATLAB 脚本: knnex-ample.m) 所示，我们已经画出了对于某些 $K=1$ 的二值数据的决策边界（每个样本的类别由与它最近的节点的类别确定）。边界的大部分看起来合理，但却有三个“岛”得出了过拟合的结果。在判别边界的错误区域中占有一个比较大的输入空间。岛中心的三个点很可能就是噪声（即错误标记点）。问题的关键是如何简单准确地递增 K 。图 5-9b 显示的是在 $K=5$ 的决策边界下的相同数据。包含更多的邻居使得边界更规则，删除了 3 个岛区域。

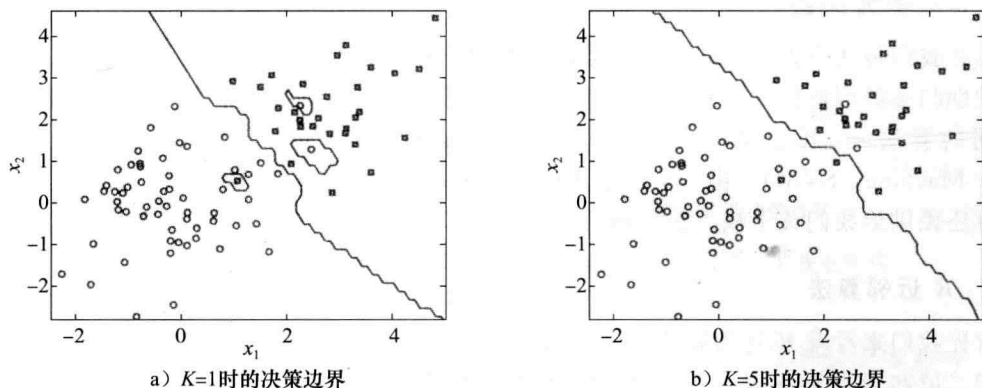


图 5-9 二值分类器数据集以及 $K=1$ 和 $K=5$ 时的决策边界

我们看到非常小的 K 值是危险的。如果 K 值过大会怎样呢？随着我们增加 K 值，我们选取距离 \mathbf{x}_{new} 更远的邻居。在某种程度上，这是非常有用的。这对于减少过拟合的可能性很有作用。然而，如果我们继续增加 K 值，那么我们将失去对数据模型的正确模式。我们考虑一个极端的例子：在某些假设的训练数据里，类别 0 和类别 1 分别有 $N_0=50$ 和 $N_1=10$ 个点。假设 $N_1=10$ ，类别 1 中的数据点至多包含 10 个邻居。因此，如果 $K \geq 21$ ，那么 \mathbf{x}_{new} 永远不会被分类为类别 1——所有区域都将被分为类别 0！我们来看图 5-10a 中一个不那么极端的例子，这里展示了一个类别 0（白色圆圈）中包含 50 个点、类别 1（灰色方块）中仅包含 20 个点的数据集。图 5-10a 中 $K=5$ 时的决策边界看起来很合理，而图 5-10b 中 $K=39$ 时的决策边界被推到了最右上角，因为这是包含更大数据集的类别发挥了它的影响。

每个类中含有的数据的点不同数据集被认为是不平衡的，在机器学习中很常见，在我们着手进行分类分析时应该对这些有所了解。5.4 节将对这个问题的更多细节进行讨论。

选择 K 比较流行的方法是交叉验证（见 1.5.2 节）。在前面的章节中，当 t 是连续的时候，我们用交叉验证的方法来优化平方损失。我们现在要对离散的（分类） t 寻找合理的方法。我们将在 5.4 节中讨论各种其他的方法，但现在我们将使用 5.2.1.6 节中使用的关于新

闻组数据的简单的方法——分类器犯错次数的比例。图 5-11b 显示的是随着图 5-11a (MATLAB 脚本: knncv.m) 中给定数据的 K 值增加, 百分误差如何改变。10 折交叉验证是为了移除 10 组数据中特定区域的影响, 整个过程重复 100 次。因此标记的错误是 $10 \times 100 = 1000$ 次错误率的平均值。随着 K 的增加, 当 $K=5$ 时分类器的错误率达到最低, 随后开始上升 (当 $K=17$ 时有一个小的回落)。

185

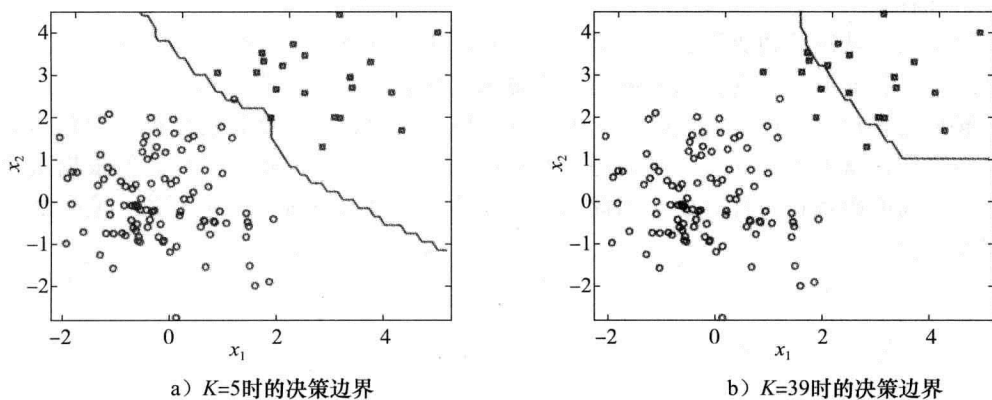


图 5-10 第二个二值分类器数据集以及 $K=5$ 和 $K=39$ 时的决策边界

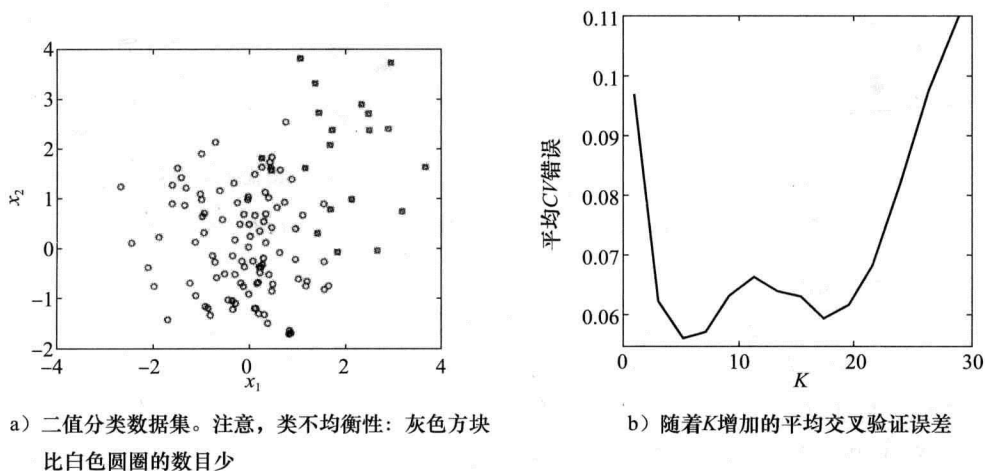


图 5-11 用交叉验证方法找到 K 的最佳取值。方法为 10 折交叉验证, 错误率为 10 个分组与 100 次对数据不同分组的平均值

5.3.2 支持向量机和其他核方法

第二个非概率分类器是支持向量机 (SVM)。这些 SVM 是二值分类器 (尽管多类扩展已经提出), 并成功应用于很多机器学习应用中。它们的成功是由于它们的出色实验表现, 并在许多应用中都很难以被打败。已经发现它们在属性个数远大于训练对象个数的应用中特别有用。这是因为 SVM 参数集的个数只与训练对象的个数有关, 而与属性个数无关。

给定 $\mathbf{w}^T \mathbf{x}_{\text{new}} + b$, 标准 SVM 使用线性决策边界来给新对象进行分类。将落在这条线一边的对象分类为 $t_{\text{new}} = 1$, 另一边的对象分类为 $t_{\text{new}} = -1$ (注意, 类别标记为 $\{1, -1\}$, 而不是 $\{0, 1\}$)。

因此, SVM 对一个新测试点 \mathbf{x}_{new} 的判定函数定义为:

$$t_{\text{new}} = \text{sign}(\mathbf{w}^T \mathbf{x}_{\text{new}} + b) \quad (5-11)$$

学习任务包括基于训练数据选择 \mathbf{w} 和 b 的值。这是通过寻找最大化间隔的值的参数来实现。这与第1章中的最小化损失、第2章中的最大化似然和第3章中的MAP方法基本上是相同的方法。

5.3.2.1 间隔

间隔定义为从决策边界到任一边最近点垂直距离。如图5-12所示，间隔被定义为 γ 。

图5-12a和b演示了为什么间隔是一个最大化的合理量值。直观地，由最大间隔构成的边界看起来更合理。而图5-12b的决策边界将会将左上和右下的点分别分类为白色类和黑色类，这与我们通常的判断相反。观察图5-12a和图5-12b中的间隔分别是如何根据决策边界和训练点之间的距离进行计算的。因此间隔定义为边界及其最邻近点的距离，它们随着边界的改变而改变。

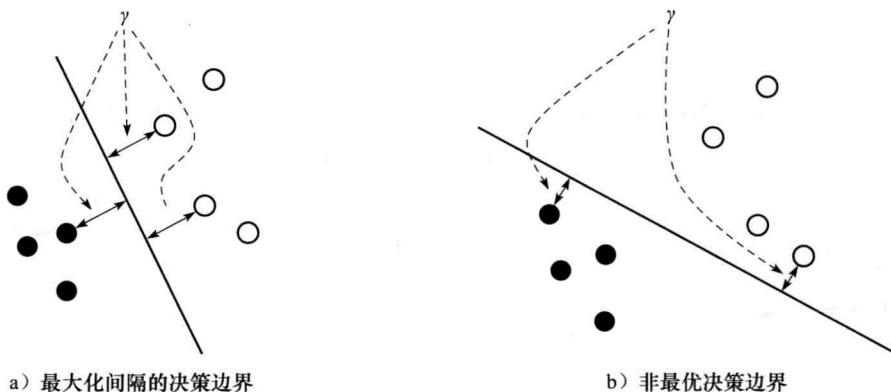


图5-12 分类器的间隔 γ ，被定义为从决策边界到任一边最近点的垂直距离

5.3.2.2 最大化间隔

可以很简单地使用每个类别中的点来计算间隔。图5-13显示了这是如何做到的。 \mathbf{x}_1 和 \mathbf{x}_2 是两个类别中距离最近的点。 2γ （即间隔两倍）等于将 \mathbf{x}_1 和 \mathbf{x}_2 在与边界垂直的方向上连接的联合向量的分量。

\mathbf{x}_1 和 \mathbf{x}_2 的联合向量定义为 $\mathbf{x}_1 - \mathbf{x}_2$ ，决策边界的垂直方向定义为 $\mathbf{w} / \|\mathbf{w}\|$ 。这两个量的内积就是我们要的值：

$$2\gamma = \frac{1}{\|\mathbf{w}\|} \mathbf{w}^T (\mathbf{x}_1 - \mathbf{x}_2)$$

我们的决策函数 $t_{\text{new}} = \text{sign}(\mathbf{w}^T \mathbf{x}_{\text{new}} + b)$ ，通过正的常数按比例缩放变量是不变的。这意味着我们可以用一个正的常数 λ 乘以 $(\mathbf{w}^T \mathbf{x}_{\text{new}} + b)$ ，而函数的输出不会改变。因此，我们可以按比例调节 \mathbf{w} 和 b ，使得 $\mathbf{w}^T \mathbf{x} + b = \pm 1$ 这两个最近的点处于边界的两边。这个约束可以简化 γ 的表达式：

$$\begin{aligned} 2\gamma &= \frac{1}{\|\mathbf{w}\|} \mathbf{w}^T (\mathbf{x}_1 - \mathbf{x}_2) \\ &= \frac{1}{\|\mathbf{w}\|} (\mathbf{w}^T \mathbf{x}_1 - \mathbf{w}^T \mathbf{x}_2) \\ &= \frac{1}{\|\mathbf{w}\|} (\mathbf{w}^T \mathbf{x}_1 + b - \mathbf{w}^T \mathbf{x}_2 - b) \end{aligned}$$

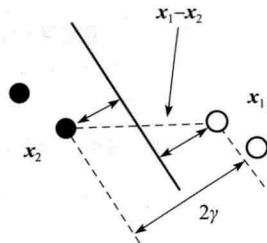


图5-13 描述了计算间隔 γ 的步骤。 2γ 等于将 $\mathbf{x}_1 - \mathbf{x}_2$ 在与边界垂直的方向上连接的联合向量的分量

$$\begin{aligned}
 &= \frac{1}{\|\mathbf{w}\|} (1+1) \\
 \gamma &= \frac{1}{\|\mathbf{w}\|}
 \end{aligned} \tag{5-12}$$

注解 5.1 (用拉格朗日乘子约束优化): 在本书不同的地方, 我们需要执行约束优化——在满足某些约束条件的情况下, 找到一系列参数值来最大化 (或最小化) 目标函数。这可以通过使用拉格朗日乘子实现。特别地, 我们构造一个新的目标函数, 使其包含每个约束的原始项和新增的项。选择这些项的形式, 使新函数的最优化等价于这个约束问题的最优化。

例如, 假设我们希望在满足 $g(\mathbf{w}) \leq a$ 的约束条件下, 最小化 $f(\mathbf{w})$:

$$\begin{aligned}
 &\underset{\mathbf{w}}{\operatorname{argmin}} \quad f(\mathbf{w}) \\
 &\text{满足 } g(\mathbf{w}) \leq a
 \end{aligned}$$

通过添加拉格朗日项 $\lambda(a - g(\mathbf{w}))$ 生成一个新的目标函数, 并同时 \mathbf{w} 和拉格朗日乘子 λ 进行最优化:

$$\begin{aligned}
 &\underset{\mathbf{w}, \lambda}{\operatorname{argmin}} \quad f(\mathbf{w}) - \lambda(g(\mathbf{w}) - a) \\
 &\text{满足 } \lambda \geq 0
 \end{aligned}$$

这里我们并不打算详细阐释该项如何起作用。在执行约束最优化时, 我们都会给出必要的拉格朗日项, 但是不会详细说明。关于这些细节, 可以参看本章结尾给出的其他阅读材料。

为了实现最大化间隔, 我们就必须最大化 $\frac{1}{\|\mathbf{w}\|}$ 。然而, 这里有一些限制。记得我们已经决定了类别 1 中的最近点, $\mathbf{w}^T \mathbf{x}_n + b = 1$ 。因此, 选择的 \mathbf{w} 必须满足所有类别 1 的点都满足 $\mathbf{w}^T \mathbf{x}_n + b \geq 1$ 。同样, 它也必须使得所有属于类别 -1 的点都满足 $\mathbf{w}^T \mathbf{x}_n + b \leq -1$ 。将标记定义为 ± 1 使我们可以将两个约束集简单地表述为:

$$t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1$$

因此, 我们的学习任务就是找到满足 N 个约束 (N 是训练集中点的个数) 的 $\gamma = \frac{1}{\|\mathbf{w}\|}$ 的最大值。实际上, 最小化 $\frac{1}{2} \|\mathbf{w}\|^2$ 更简单, 因此我们以计算实现。从形式上, 最优化问题就变为:

$$\begin{aligned}
 &\underset{\mathbf{w}}{\operatorname{argmin}} \quad \frac{1}{2} \|\mathbf{w}\|^2 \\
 &\text{满足 } t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1 \quad \text{对于所有的 } n
 \end{aligned}$$

这是我们第一次遇到有约束的最优化问题。为了解决它, 我们需要通过拉格朗日乘子将约束加入到目标函数中。拉格朗日乘子给目标函数的每个约束增加一项, 这样使得新目标函数的最优化等价于原始约束问题的最优化。在我们的例子中, 我们需要 N 个拉格朗日项。每项都关联一个拉格朗日乘子, 并限定为整数。不需要考虑拉格朗日更多的细节, 我们的新目标函数为:

$$\begin{aligned}
 &\underset{\mathbf{w}, \alpha}{\operatorname{argmin}} \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{n=1}^N \alpha_n (t_n(\mathbf{w}^T \mathbf{x}_n + b) - 1) \\
 &\text{满足 } \alpha_n \geq 0 \quad \text{对于所有的 } n
 \end{aligned}$$

这里我们用到 $\|\mathbf{w}\|^2 = \mathbf{w}^T \mathbf{w}$ 。在目标函数 \mathbf{w} 和 b 的偏导数取 0 的时候, 这个新目标函数取得最优解。这些导数为:

$$\frac{\partial}{\partial \mathbf{w}} = \mathbf{w} - \sum_{n=1}^N \alpha_n t_n \mathbf{x}_n$$

$$\frac{\partial}{\partial b} = - \sum_{n=1}^N \alpha_n t_n$$

189 令这两个表达式等于0, 可以得出满足最优化的两个表达式:

$$\mathbf{w} = \sum_{n=1}^N \alpha_n t_n \mathbf{x}_n \quad (5-13)$$

$$\sum_{n=1}^N \alpha_n t_n = 0 \quad (5-14)$$

将这两个表达式代入到目标函数, 可以得到关于 α_n 而不是 \mathbf{w} 最大化的新目标函数:

$$\begin{aligned} & \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{n=1}^N \alpha_n (t_n (\mathbf{w}^T \mathbf{x}_n + b) - 1) \\ &= \frac{1}{2} \left(\sum_{m=1}^N \alpha_m t_m \mathbf{x}_m^T \right) \left(\sum_{n=1}^N \alpha_n t_n \mathbf{x}_n \right) - \sum_{n=1}^N \alpha_n \left(t_n \left(\sum_{m=1}^N \alpha_m t_m \mathbf{x}_m^T \mathbf{x}_n + b \right) - 1 \right) \\ &= \frac{1}{2} \sum_{n,m=1}^N \alpha_m \alpha_n t_m t_n \mathbf{x}_m^T \mathbf{x}_n - \sum_{n,m=1}^N \alpha_m \alpha_n t_m t_n \mathbf{x}_m^T \mathbf{x}_n - \sum_{n=1}^N \alpha_n t_n b + \sum_{n=1}^N \alpha_n \\ &= \sum_{n=1}^N \alpha_n - \frac{1}{2} \sum_{n,m=1}^N \alpha_m \alpha_n t_m t_n \mathbf{x}_m^T \mathbf{x}_n \end{aligned}$$

这里, 我们利用 $\sum_{n=1}^N \alpha_n t_n = 0$ 删除了倒数第二行的第三项。这个表达式被认为是一个双重优化问题, 并且根据下面的约束进行最大化:

$$\alpha_n \geq 0, \quad \sum_{n=1}^N \alpha_n t_n = 0$$

第二个约束来自式 (5-14)。注意, \mathbf{w} 在这个最优化问题中已经不起作用。

这个最优化问题是一个有约束的二次方程问题, 因为 $\alpha_n \alpha_m$ 项导致其为二次方程。对此还没有分析方法但可以合理简单数字化地解决。例如, Matlab 的函数 quadprog 可以解决这类问题。

5.3.2.3 预测

假设给定一个最佳的 α_n 集合, 我们如何做出预测? 我们的决策函数 $t_{\text{new}} = \text{sign}(\mathbf{w}^T \mathbf{x}_{\text{new}} + b)$ 基于 \mathbf{w} 和 b , 不基于 α_n 。为了将它转变为 α_n 的函数, 我们用式 (5-13) 中的表达式代替 \mathbf{w} , 得到:

$$t_{\text{new}} = \text{sign} \left(\sum_{n=1}^N \alpha_n t_n \mathbf{x}_n^T \mathbf{x}_{\text{new}} + b \right) \quad (5-15)$$

190 为了找到 b , 我们将利用 $t_n (\mathbf{w}^T \mathbf{x}_n + b) = 1$ 。将式 (5-14) 代入表达式并重新排列, 可以得出 b (注意, $1/t_n = t_n$):

$$b = t_n - \sum_{m=1}^N \alpha_m t_m \mathbf{x}_m^T \mathbf{x}_n \quad (5-16)$$

其中, \mathbf{x}_n 为距离边界最近的点。现在我们得到分类 \mathbf{x}_{new} 需要的所有条件。

5.3.2.4 支持向量

距离最大化间隔决策边界最近点的集合称为支持向量。用这个名字, 因为这些点定义或者支持这个决策边界。由于决策边界由最大化间隔得到, 并且边缘只取决于那些距离最近的点, 所以我们可以去掉其他所有的数据, 只留下决策边界。在取得最优化时, 所有的 α_n 都将取零。如果它们是非零的, 那么它们将会对决策边界产生影响 (见式 (5-15))。在许多应用中, 这是一个很简单的方法——决策仅仅是训练例子的一个小子集的函数。面对更大数据的问题, 这将是一个很有用的特点。假设当存在数千个训练对象的数据, 考虑用 KNN 对测

试点进行分类。为了找到邻居集，我们不得不计算新对象与所有测试对象的距离。而对于在同样数据集上训练的 SVM，决策函数仅仅包括训练数据的一个小子集。

图 5-14 显示了一个二值数据集和利用支持向量（灰色大圆圈）求得的决策边界（当 $\mathbf{w} = \sum_{n=1}^N \alpha_n t_n \mathbf{x}_n$ 时， $\mathbf{w}^T \mathbf{x} + b = 0$ ）。它们都是 $\alpha_n > 0$ 且分类新数据唯一能用到的点。

尽管只使用 3 个训练点就做出判断被认为很高效的，但这并不总是好事。图 5-15（MATLAB 脚本：svmhard.m）阐述了原因。这里我们看到的数据与图 5-14 中的数据一样，只有一点不同——属于灰色方块的支持向量与另一种类别更接近。移动这个单个数据点对整个决策边界的位置有很大影响。这是另一个过拟合的例子——我们让这个数据起到了太大的作用。为了解它发生的原因，我们需要看最原始的约束：

$$t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1 \quad (5-17)$$

这意味着所有的训练点都必须处于决策边界正确的一边。这类 SVM 称为硬间隔 SVM。有时候放宽这个约束会更合理一点（会有更好的泛化表现）。幸运的是，用软间隔会更简单直接。

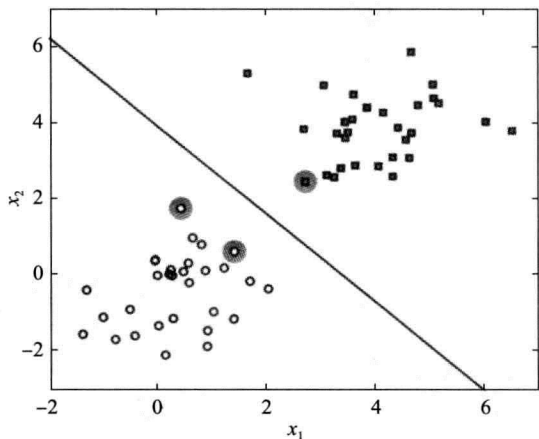


图 5-14 线性 SVM 的决策边界和支持向量

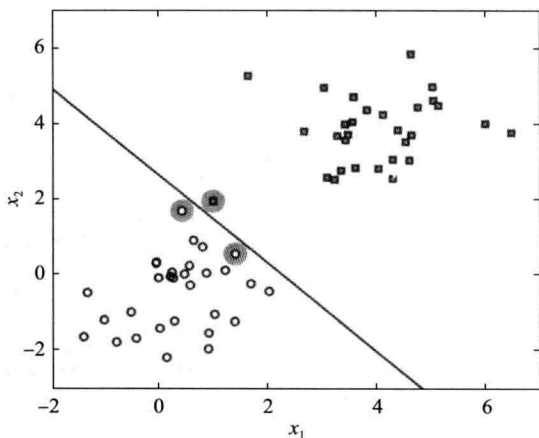


图 5-15 一个线性 SVM 的决策边界和支持向量。灰色方块类的支持向量显然发挥了太多的作用

5.3.2.5 软间隔

为了允许点可能处于边界错误的一边，我们需要放宽原始公式的约束。尤其需要放宽式 (5-17) 的约束以便允许有些点落在决策边界的间隔（或者落在错误的一边）。为了满足这些，约束变为：

$$t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1 - \xi_n \quad (5-18)$$

其中 $\xi_n \geq 0$ 。如果 $0 \leq \xi_n \leq 1$ ，那么点位于边界正确的一边，或者落在边界上。如果 $\xi_n > 1$ ，那么点就位于边界错误的一边。最优化任务变为：

$$\operatorname{argmin}_{\mathbf{w}} \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N \xi_n$$

满足 $\xi_n \geq 0$ 和 $t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1 - \xi_n$ 对于所有的 n

新的参数 C 控制点位于间隔带或者决策边界错误一边的最大距离。如果我们继续使用在硬间隔分类中的步骤，就会发现模型的这个改变对于最大化问题只有很小的影响。省略具体细节（见练习 EX 5.8），我们现在需要找到下面二次方程问题的最大值：

$$\operatorname{argmax}_{\mathbf{w}} \sum_{n=1}^N \alpha_n - \frac{1}{2} \sum_{n,m=1}^N \alpha_n \alpha_m t_n t_m \mathbf{x}_n^T \mathbf{x}_m$$

191

192

满足 $\sum_{n=1}^N \alpha_n t_n = 0$ 和 $0 \leq \alpha_n \leq C$, 对于所有的 n

唯一不同的就是 α_n 的上界 (C)。每个训练点对于决策函数的影响都是与 α_n 成正比的。因此, 我们给每个训练点的影响都增加一个上界。如图 5-15 中的例子, 灰色类的支持向量有 $\alpha_n = 5.45$ 。设置 C 为 1, 将会引起决策边界的改变 (有些灰色方块类点的 α_n 将变为非零), 它会向灰色方块类的其他对象方向移动。图 5-16 显示了当 $C=1$ 和 $C=0.01$ 时的决策边界和支持向量 (MATLAB 脚本: `svnssoft.m`)。随着 C 的增加, 每个训练点对最大值的潜在影响被削弱, 因此越来越多的点在决策函数中起作用。

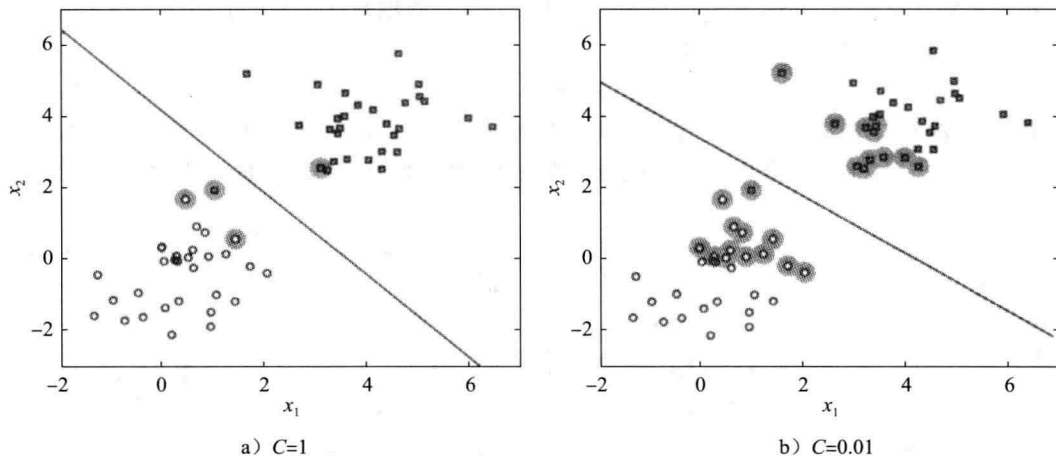


图 5-16 软间隔中参数 C 的两个不同取值的决策边界和支持向量。离群支持向量的影响被削弱了

使用软间隔, 我们需要固定参数 C 。类似于 KNN 中的 K , 我们可以利用交叉验证来确定它。它的方法步骤和误差估计与 KNN 是相同的, 所以这里我们就省略细节了。最后的一点就是 b 的计算。我们不再使用支持向量来计算它, 因为它们根本不满足 $t_n(\mathbf{w}^T \mathbf{x}_n + b) = 1$ 。在边缘带中 (或者错误一边) 的支持向量有 $t_n(\mathbf{w}^T \mathbf{x}_n + b) < 1$ 。为了解决这个问题, 应该找到 $\mathbf{w}^T \mathbf{x}_n$ (或 $\sum_m \alpha_m t_m \mathbf{x}_m^T \mathbf{x}_n$) 值最大的支持向量, 并利用式 (5-16) 计算 b 。

5.3.2.6 核

至此, 我们对 SVM 的研究始终局限于线性决策边界。软间隔允许训练点落在决策边界错误的一边, 但如果数据如图 5-17 所示的那样复杂, 这就起不到作用了。如果我们想要一个非线性函数, 就需要给 \mathbf{x} 加上一些项, 并扩展 \mathbf{w} 。用 SVM, 我们采用了一个非常不同的方法。模型 (线性决策边界) 还是一样, 而是将数据转换到一个新的空间中。转换是为了将转换后的数据可以被线性决策边界分类。

为了阐明这个观点, 考虑图 5-17 中的数据。这些数据无法被一条直线分开。但是如果用 $\mathbf{x}_n = \{x_{n1}, x_{n2}\}^T$ 来代替每个数据点, 用 $z_n = x_{n1}^2 + x_{n2}^2$ 代替它们与源点的距离, 就可以用一条直线来分离它们: 圆圈类的点距离源点的距离都比方块类的点远。在 SVM 中用 z_n 取代 \mathbf{x}_n , 就根

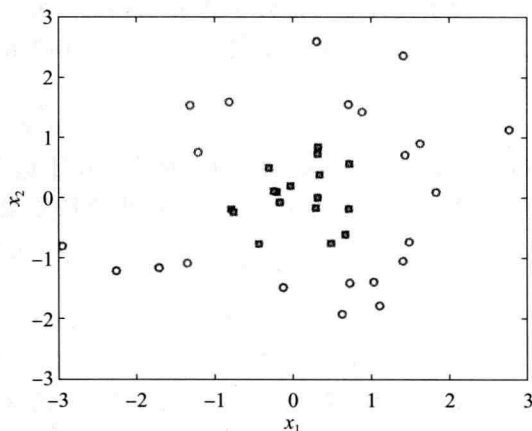


图 5-17 线性决策边界无法合理分类的二值数据集

本不需要重新定义算法。当我们有一个测试点 \mathbf{x}_{new} 时, 我们只需要计算 z_{new} , 并用普通的方法来对它进行分类。通常, 我们用 $\phi(\mathbf{x}_{\text{new}})$ 定义第 n 个训练对象的转换。

也许 SVM 构架最重要的特点就是从不需要执行转换。在我们的目标和决策函数中, 数据 \mathbf{x}_n 、 \mathbf{x}_m 、 \mathbf{x}_{new} 变为它们的内积: $\mathbf{x}_n^T \mathbf{x}_m$, $\mathbf{x}_n^T \mathbf{x}_{\text{new}}$ 等。我们从来看不到 \mathbf{x} 以它的本身出现。在转换后, 在新的空间中计算内积: $\phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m)$ 。首先需将每个数据点进行特定的转换, 然后在新空间中计算内积。然而, 我们并不需要考虑转换的项。相反, 如果我们对某些转换 $\phi(\cdot)$ 构造函数 $k(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m)$, 我们就可以在表达式中用 $k(\mathbf{x}_n, \mathbf{x}_m)$ 代替原始空间中的内积。在某些空间中与内积有关的函数称为核函数。

用核函数重新定义最优化和决策函数(软边缘版)为:

$$\arg\max_w \sum_{n=1}^N \alpha_n - \frac{1}{2} \sum_{n=1}^N \alpha_n \alpha_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

$$\text{满足 } \sum_{n=1}^N \alpha_n t_n = 0 \text{ 和 } 0 \leq \alpha_n \leq C, \text{ 对于所有的 } n$$

$$t_{\text{new}} = \text{sign}\left(\sum_{n=1}^N \alpha_n t_n k(\mathbf{x}_n, \mathbf{x}_{\text{new}}) + b\right)$$

各种各样的核函数(每个核函数都相当于某种转换的一个内积)我们都可以使用。下面是三种最常用的核函数:

$$\text{线性核函数 } k(\mathbf{x}_n, \mathbf{x}_m) = \mathbf{x}_n^T \mathbf{x}_m$$

$$\text{高斯函数 } k(\mathbf{x}_n, \mathbf{x}_m) = \exp\{-\gamma(\mathbf{x}_n - \mathbf{x}_m)^T(\mathbf{x}_n - \mathbf{x}_m)\} \quad (5-19)$$

$$\text{多项式函数 } k(\mathbf{x}_n, \mathbf{x}_m) = (1 + \mathbf{x}_n^T \mathbf{x}_m)^\gamma$$

线性核相当于我们使用的 SVM。高斯核和多项式核更具有灵活性, 而且都有额外的用户定义的参数(γ)——通常经过交叉验证得到。

图 5-18 是图 5-17 中的数据当 $\gamma=1$ 、 $C=10$ (所有的 $\alpha_n < C$, 所以这实际上是一个硬边缘)时用高斯核计算得出的结果(MATLAB 脚本: svmgauss.m)。这个决策边界看起来很合理。对于原始的 SVM, 我们可以计算出由满足:

$$\mathbf{w}^T \mathbf{x} + b = 0$$

的 \mathbf{x} 组成的决策边界。我们不用再计算 \mathbf{w} , 因为它已经由 $\sum_n \alpha_n t_n \phi(\mathbf{x}_n)$ 给出, 并且我们也不需要知道 $\phi(\mathbf{x}_n)$ (我们只知道 $k(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m)$)。因此, 为了画出决策边界, 我们必须根据 \mathbf{x}_{new} 的坐标值估算 $\sum_n \alpha_n t_n k(\mathbf{x}_n, \mathbf{x}_{\text{new}})$, 然后根据 $\sum_n \alpha_n t_n k(\mathbf{x}_n, \mathbf{x}_{\text{new}}) = 0$ 利用 Matlab 画出轮廓。

如果我们改变 γ 会怎样? 修改 γ 会改变转换 $\phi(\mathbf{x}_n)$, 进而改变在原始域中我们想看到的决策边界的类型(它们在转换域中是线性的)。在高斯核中, 增加 γ 会增加原始域中决策边界的复杂性。当分别用 $\gamma=0.01$ 和 $\gamma=50$ 来比较图 5-19a、b (MATLAB 脚本: svmgauss.m) 时, 结果就很明显了。在图 5-19a 中, 决策边界过于简单——它不能在原始域中足够迅速地弯曲从而仅仅围住方块类的数据。相反, 当 $\gamma=50$ (图 5-19b) 时, 决策边界太灵活, 使得它看起来过于复杂。这两种方案中, 值得注意的是支持向量的数量急剧增长

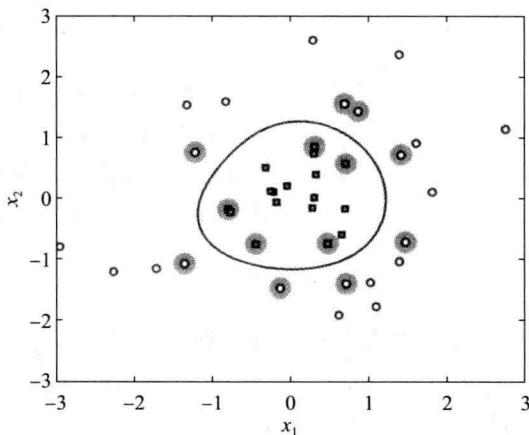


图 5-18 用图 5-17 中的数据, 当核参数 $\gamma=1$ 、 $C=10$ 时用高斯核计算出的决策边界和支持向量

(参见图 5-18), 解决办法不再认为那么少了。

这个模型复杂度问题与我们在第 1 章遇到的问题是一样的。那里我们发现超过某一个固定点, 增加多项式的次数会导致奥运会 100 米模型出现不好的预测结果。这里, 模型太简单(图 5-19a) 或者太复杂(图 5-19b) 都会产生不好的预测结果。在过于简单的例子中, 模型过于频繁地预测出灰色方块类, 而在过于复杂的例子中, 就恰恰相反。正如第 1 章中提到的, 我们必须很仔细地设置 γ , 以便让复杂性适中, 例如利用交叉验证方法。让问题更糟的是, 参数 C 和 γ 会协同作用于模型。我们不能最优化一个, 再最优化另一个; 我们必须两个同时做。当训练集很大的时候 (N 很大) 这相当成问题。SVM 可以解决一个 N 维最优化问题。对于一个很大的 N , 这会非常费时, 而且寻找两个参数 (C, γ) 的交叉验证将会执行最优化过程很多遍。

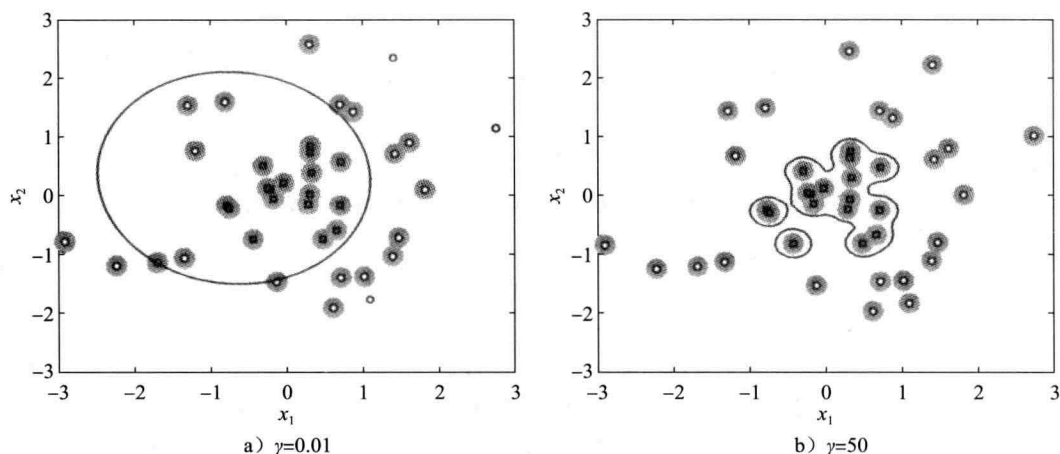


图 5-19 用图 5-17 中的数据, 当核参数 $C=10$ 时, 不同的 γ 值用高斯核计算出的决策边界和支持向量

SVM 并不是唯一可以核化的算法。许多机器学习算法都可以表述为这种形式, 数据都以内积形式存在。这意味着在许多算法中, 我们不增加算法的复杂性, 就可以解决复杂的问题(例如, 高度拟合非线性决策边界)。我们将在第 6 章研究聚类时看到另一个例子。我们还可以核化另一个非概率分类器, KNN。KNN 需要计算每个 \mathbf{x}_n 和 \mathbf{x}_{new} 之间的距离。这个距离可以表示为:

$$(\mathbf{x}_{\text{new}} - \mathbf{x}_n)^T (\mathbf{x}_{\text{new}} - \mathbf{x}_n)$$

如果我们把它乘开, 就得到了内积:

$$\mathbf{x}_{\text{new}}^T \mathbf{x}_{\text{new}} - 2\mathbf{x}_{\text{new}}^T \mathbf{x}_n + \mathbf{x}_n^T \mathbf{x}_n$$

用核化公式取代之后,

$$k(\mathbf{x}_{\text{new}}, \mathbf{x}_{\text{new}}) - 2k(\mathbf{x}_{\text{new}}, \mathbf{x}_n) + k(\mathbf{x}_n, \mathbf{x}_n)$$

便得到了一个核化的 KNN。

5.3.3 小结

在前面的章节中, 我们描述了 4 个流行的分类算法, 并叙述了它们的用法。这 4 个算法为我们之后的学习奠定了坚实的基础, 使我们可以用数据进行分类试验, 并继续探索其他的分类技术。

可以提供一个特定的算法, 仅仅是分类分析的一个部分。另一个重要应用就是如何分析一个分类器的表现, 这将在 5.4 节中重点讲述。

5.4 评价分类器的性能

在接下来的讨论中,假设我们要对 N 个相互独立的测试点 $\mathbf{x}_1, \dots, \mathbf{x}_n$ 的分类结果进行性能评价,已知 N 个点的类别标号分别为 t_1^*, \dots, t_n^* , 分类器预测出的类别标号为 t_1, \dots, t_n 。它们是完全独立的数据集或者是一个交叉验证集中的数据。

5.4.1 准确率——0/1 损失

当我们需要表述对性能的评估时,我们采用到分类器的准确性,即 0/1 损失。至于为什么用这个名字,因为对一个特定的测试点,损失或者为 0 或者为 1,取决于预测是正确的 ($t_n = t_n^*$) 还是错误的 ($t_n \neq t_n^*$)。当我们求测试集中 N 个对象的平均值时,这个量显示的是分类器错误分类的个数。这个数可以被解释为一个随机测试点被错误分类的可能性。显然,这个值越低越好。尽管这个方法被广泛应用,但它还是有一些不足之处。特别地,如何评价这个量并不总是很容易——例如,0.2 怎么样? 考虑两个假设的二值分类器的问题: 第一个,我们观察每种类别个数相同的数据; 第二个,类别 1 中我们取 80%, 类别 2 中我们取 20%。在第一个例子中,0/1 损失为 0.2 的平均值可能说明性能还不错。而第二个就不见得了。我们总是将对象归为类别 1,却也能得到 0.2 的平均损失。因此,对于类别数据不平衡的数据,运用 0/1 损失的时候应该非常小心。现在我们将介绍一个能够克服这个问题的方法。

5.4.2 敏感性和特异性

想象一个探测疾病的二值分类任务。 $t=0$ 代表健康的人, $t=1$ 代表患病的人。如果我们试图去检测一种稀有的疾病,那么 0/1 损失就是一个糟糕的主意——诊断所有人都健康就能得到非常低的 0/1 损失。分析敏感性和特异性这两个值,是一个更好的主意。要计算敏感性和特异性,它们需要从分类结果中提取 4 个统计值。它们是:

- 正确的正例 (TP) —— 本身为类别 $t_n^* = 1$, 被分类为 $t_n = 1$ 的对象的数量 (患病的人被诊断为患病)。
- 正确的反例 (TN) —— 本身为类别 $t_n^* = 0$, 被分类为 $t_n = 0$ 的对象的数量 (健康的人被诊断为健康)。
- 错误的正例 (FP) —— 本身为类别 $t_n^* = 0$, 被分类为 $t_n = 1$ 的对象的数量 (健康的人被诊断为患病)。
- 错误的反例 (FN) —— 本身为类别 $t_n^* = 1$, 被分类为 $t_n = 0$ 的对象的数量 (患病的人被诊断为健康)。

得到这些值,我们就可以计算敏感性:

198

$$S_e = \frac{TP}{TP + FN} \quad (5-20)$$

特异性为:

$$S_p = \frac{TN}{TN + FP} \quad (5-21)$$

两个值都位于 0 和 1 之间。

一般来说,这两个值分别告诉我们如何善于诊断出患病的人和健康的人。敏感性是被正确诊断为患病的人 (TP) 与所有患病的人 (TP+FN) 的比。特异性是被正确诊断为健康的人 (TN) 与所有健康的人 (TN+FP) 的比。

思考这个稀有疾病的例子，如果我们判断所有人都健康，那么我们得到的敏感性将是 1（非常好，我们正确诊断了所有健康的人），但是特异性却是 0（我们误诊了所有患病的人），这非常糟糕。理想情况下，我们希望 $S_e = S_p = 1$ ——完美的敏感性和特异性。这对于所有的应用不并一定合适，我们需要定义如何最优化敏感性和特异性的值。例如， $S_p = 0.9$ 、 $S_e = 0.8$ 和 $S_p = 0.8$ 、 $S_e = 0.9$ 哪个更好？这个答案依赖于问题。在我们稀有疾病诊断中，我们不想漏诊任何一个患病的人，但可以容忍将健康的人诊断为患病（他们可以通过更多的测试，稍后被发现是健康的）。由此，我们可能希望减小 S_p 从而提高 S_e 。在其他应用中，我们可能采取相反的方法。

通常我们将敏感性和特异性设置为一个固定的值会很方便。这可以通过评价接收者操作特征（Receiver Operating Characteristic, ROC）曲线下的区域得到。

5.4.3 ROC 曲线下的区域

在许多分类算法中，我们都提供一个实数值的输出，从而根据阈值来进行分类。例如，在贝叶斯分类器（二值）和逻辑回归中，我们提供了 $P(T_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$ ——值位于 0 和 1 之间。在 SVM 中，我们提供了以 0 为阈值（通过一个标记函数）的函数。对于任何算法，我们都可以使用任意阈值来获得一个硬分类器。例如，我们可以规定如果 $P(T_n = 1 | \mathbf{x}_n, \mathbf{X}, \mathbf{t}) > 0.7$ ，那么 \mathbf{x}_n 就应该属于类别 1。在 SVM 中，我们可以将阈值定为 0.2 而不是 0，从而使得 \mathbf{x}_n 更不容易被分类为类别 1。

接收者操作特征（ROC）曲线让我们可以观察随着阈值的改变，性能是如何变化的。由一系列阈值计算得出了敏感性和特异性，描绘了敏感性和特异性或者假阳率（ $1 - S_p$ ）如图 5-20 所示（MATLAB 脚本：svmroc.m）。这些曲线是由图 5-19a 和图 5-19b 中太简单和太复杂的模型用 1000 个独立测试集计算得到的。我们知道我们想让 S_e 和 S_p 尽可能地高。因此，曲线越接近左顶点（ $S_e = 1, 1 - S_p = 0$ ）越好。如果曲线达到左顶点，说明我们可以选择一个可以完美分类数据的阈值。这条曲线总会从 $S_e = 0, 1 - S_p = 0$ 处开始，意味着这个阈值永远不会把数据分类为类别 1，并且在 $S_e = 1, 1 - S_p = 1$ 处结束，意味着分类器永远不会将数据分类为类别 0（在 SVM 中是 -1）。随着分类器变差，曲线将逐渐趋向于一条从 (0, 0) 到 (1, 1) 的直线。这相当于随机进行分类。

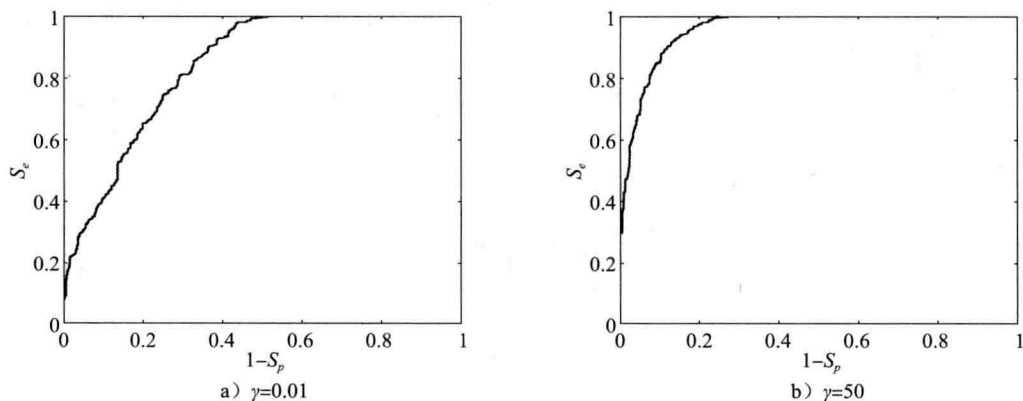


图 5-20 图 5-19a、b 中的 SVM 的 ROC 曲线

基于图 5-20 的显示， $\gamma = 50$ 的 SVM 更接近于左顶点，因此它比 $\gamma = 0.01$ 好。我们可以通过计算 ROC 曲线下方的区域（称为 AUC）来评估性能。一个能够完美分类数据的分类器应该有大小为 1 的 AUC（曲线沿着左手边垂直升到左顶点，在径直穿过顶边）。随机分类的

分类器的 AUC 是 0.5（正如上面提到的，曲线变为一条从 (0, 0) 到 (1, 1) 的直线）。图 5-20 中曲线的 AUC 分别是 0.8348 和 0.9551。在图 5-21 中，我们绘制了当 $\gamma=1$ （图 5-18 中的 SVM）时的 ROC 曲线。这个例子中的 AUC 为 0.9936——正如我们期望的，它是三个中最好的。

在大多数应用中，AUC 是一个比 0/1 损失更好的评价性能的方法。它利用敏感性和特异性来考虑类别数据的不平衡性。它的一个缺点就是无法扩展到多值分类器。把它用在多值分类器中的一个办法就是以多个二值问题的形式来分析分类器的结果。例如，如果我们有个 3 个类别，我们可以做 3 个 ROC，每个 ROC 都是考虑是否为类别 c 的二值问题。这将会提供每个分类器有用的信息，但是如何将 3 个 AUC 值组合起来却并不清楚。我们现在看看最后一个能够简单（并且非常有效）扩展到多值分类器的性能分析工具。

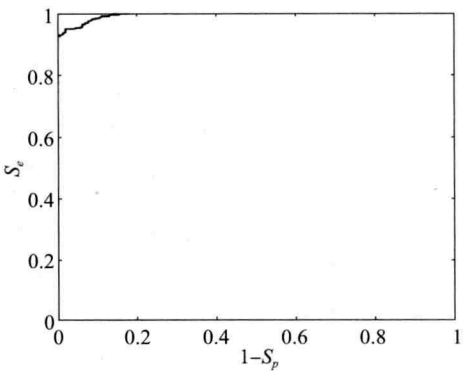


图 5-21 图 5-18 中 SVM 的 ROC 曲线。这个曲线很难看见，因为对于大多数阈值 $S_c=1$ 或者 $1-S_p=0$

200

5.4.4 混淆矩阵

5.4.2 节中介绍的 4 个值（TP、TN、FP、FN）经常在一个表中出现。在一个有两种类别的例子中，这个表会有两行和两列。行代表预测的类别 (t)，列代表实际的类别 (t^*)。表 5-2 显示的就是这样的表，即混淆矩阵。二值问题的混淆矩阵仅仅描述 4 个值。然而，混淆矩阵真正发挥作用的是在多类问题中。一个含有 20 个类新闻组数据的混淆矩阵有 20 行和 20 列，并且可以让我们发掘算法的好坏，如表 5-3 所示。乍看起来，它可能很复杂、不好用，但它能合理直接地提取有用的信息。例如，对角线上的大值告诉我们，整体上，分类器性能很好。非对角线上比较大的元素说明对数据做出了规律性的错误分类。例如，68 个原本属于类别 20 的元素被错误地分类为类别 16——我们已经在 5.2.1.6 节中讨论过的现象。同样，大量属于类别 19 的元素被错误地分类为类别 17。这样的分析不仅让我们发现错误在哪里（如果我们简单计算 0/1 损失就无法得到这些细节），而且给我们提供如何提高性能的建议。在 5.2.1.6 节中，我们看到类别 20 和类别 16 非常相似，类别 19 和类别 17 也是。也许区分它们本身就很困难。类别 20 最容易被错误分类，所以如果我们想提高性能，我们就应该考虑——收集更多的信息或者将它与其他类别合并。

200
}
201

表 5-2 二值混淆矩阵

		真类	
		1	0
预测类	1	TP	FP
	0	FN	TN

表 5-3 20 个类新闻组数据的混淆矩阵

		真类																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
预测类	1	242	3	3	0	1	0	0	1	0	4	2	0	2	10	4	7	1	12	7	47
	2	0	296	33	8	8	42	9	1	1	0	0	4	18	7	8	2	0	1	1	3
	3	0	6	209	15	9	8	4	0	0	0	0	1	0	1	0	1	0	0	0	0

(续)

		真类																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
预 测 类	4	0	12	60	303	36	12	46	2	0	1	0	1	28	3	0	0	0	0	0	0
	5	0	8	10	22	277	2	21	0	0	1	0	2	7	0	0	1	1	0	0	0
	6	1	21	30	2	2	304	0	1	0	3	0	1	3	0	1	2	0	0	1	0
	7	0	1	0	5	5	1	235	5	1	2	0	1	1	0	0	0	1	0	0	0
	8	0	3	1	6	4	0	31	356	25	3	1	0	9	4	0	0	2	2	1	0
	9	0	2	2	0	1	2	5	4	353	1	0	0	2	0	1	0	1	1	0	1
	10	0	0	2	0	1	1	0	2	2	348	4	0	0	1	0	0	1	1	0	0
	11	1	0	1	1	0	0	1	0	0	16	382	0	1	0	1	0	1	1	0	0
	12	1	16	16	5	4	10	3	1	1	2	0	360	45	0	4	1	3	4	3	1
	13	1	4	1	24	16	0	9	5	1	2	0	3	260	3	4	0	0	0	0	0
	14	2	3	4	0	8	0	2	0	1	0	2	2	6	324	4	1	1	0	3	3
	15	3	7	4	1	2	3	3	2	0	0	1	0	4	3	336	0	2	0	7	5
	16	39	4	5	0	0	1	3	1	1	3	2	2	5	17	4	376	3	7	2	68
	17	4	0	0	0	3	1	1	5	4	1	0	9	0	3	1	3	325	3	95	19
	18	7	1	0	0	0	1	3	1	2	2	1	0	2	6	2	1	2	325	4	5
	19	7	2	9	0	6	2	5	8	5	8	4	8	0	10	21	1	16	19	185	7
	20	10	0	1	0	0	0	1	0	0	0	0	1	0	1	1	2	4	0	1	92

202

5.5 判别式和产生式分类器

在我们的讨论中，我们将分类器分为概率方法的和非概率方法的。另一个常见的方法是根据分类器是判别式还是产生式来进行分类。产生式分类器为每个类别定义一个模型，然后将新元素指定为最适合它们的模型。另一方面，判别式分类器清楚地定义类别之间的决策边界。贝叶斯分类器（5.2.1节）是产生式分类器的例子，而SVM（5.3.2节）和逻辑回归（5.2.2节）是判别式分类器的例子。

5.6 小结

本章介绍了4种流行的分类算法——两种提供概率输出，两种提供硬分类。在现有的领域中我们无法公正地评价它们——本书是用SVM和其他单独的核函数写。然而，现有的资料应该足够读者使用和进行实验。另外，本章也提供了关于分类整体问题和各种类型分类算法足够的背景知识以便读者能够发掘其他算法并将之归类。

除了描述算法外，我们还关注了如何评价分类器的好坏以及一些我们可能遇到的问题。然而，我们紧紧接触了表面。还有许多不同领域的性能评价方法和许多我们可能遇到的问题。

5.7 练习

EX 5.1 假设对于所有的类有 $\Sigma_c = I$ ，对于贝叶斯分类器的某一 μ_c 计算其后验密度 $p(\mu_c | X^c)$ ，其中第 c 类的训练数据由 x_1, \dots, x_{N_c} 表示。假设 $p(\mu_c)$ 服从高斯分布。

EX 5.2 使用上一题中计算出的后验概率计算期望似然函数

$$p(x_{\text{new}} | T_{\text{new}} = c, X, t) = E_{p(\mu_c, \Sigma_c | X^c)} \{ p(x_{\text{new}} | \mu_c, \Sigma_c) \}$$

EX 5.3 计算贝叶斯分类器的第 c 类的参数 μ_c 与 Σ_c 的最大似然估计，假设其服从高斯类条件分布且第 c 类的训练数据由 x_1, \dots, x_{N_c} 表示。

203

EX 5.4 计算贝叶斯分类器的第 c 类的参数 q_{mc} 的最大似然估计, 假设其服从多项式类条件分布且 N_c 中的 M 维对象由 $\mathbf{x}_1, \dots, \mathbf{x}_{N_c}$ 表示。

EX 5.5 对于一个具有多项式类条件分布的贝叶斯分类器, 其参数 \mathbf{q}_c 具有 M 维, 当 \mathbf{q}_c 的先验分布为参数为常数 α 的狄利克雷分布时, 计算第 c 类的后验狄利克雷, 第 c 类的训练数据由 $\mathbf{x}_1, \dots, \mathbf{x}_{N_c}$ 表示。

EX 5.6 使用上一题中计算出的后验概率计算期望似然函数

$$p(\mathbf{x}_{\text{new}} | T_{\text{new}} = c, \mathbf{X}, t) = E_{p(\mathbf{q}_c | \mathbf{X}^c)} \{ p(\mathbf{x}_{\text{new}} | \mathbf{q}_c) \}$$

EX 5.7 使用 EX 5.4 中计算出的结果计算 q_{cm} 的 MAP 估计。

EX 5.8 简述软间隔 SVM 中的双重优化问题。

204

其他阅读材料

- [1] Ken Binmore and Joan Davies. *Calculus: Concepts and Methods*. Cambridge University Press, 2002.

优化问题中有关拉格朗日使用方法的很好描述。

- [2] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

支持向量机及其他核方法的全面介绍。

- [3] Richard Duda, Peter Hart, and David Stork. *Pattern Classification*. Wiley-Interscience, second edition, 2000.

以分类为主题的一本教程。

- [4] T. Furey et al. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

把支持向量机应用到高维微阵列数据中的论文之一, 也描述了简单的特征提取方法。

- [5] Brian Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

20 世纪 90 年代末的一本经典的模式识别教程。

- [6] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

简洁的核方法介绍以及许多应用的例子。与该作者其他同类书籍相比, 该书的介绍更广泛、结合的例子更实际。

205

聚类分析

到目前为止，我们一直关注**有监督学习**。在所有的任务中，我们得到的数据是一组数据对象及其相关的标签（或目标）。例如，由奥运会举办年所构成的对象和相应百米比赛获胜时间的目标；由文档组成的对象及其类别组成的目标。目标 t_n 的存在使得这些任务得到了监督。

有时我们得到的对象是无标签的。分析这种数据需要使用**无监督机器学习**技术。初看起来，也许很难理解用这样的数据能够做什么。当然，如果只知道现代奥运会的举办年，的确没有太多事情可以做。

在本章和第7章中，我们将学习两种广泛使用的，用于这种无监督情况的机器学习技术。本章介绍第一种：聚类（clustering），我们具体将关注两种聚类方法——K 均值（K-means）和混合模型（mixture models）。

6.1 一般问题

聚类分析的目标是，创建满足处于同一组内的对象相似、不同组的对象相异的对象分组。有很多定义两个对象相似的方法。在相似度定义的基础上，也有多种进行分组的方法。在详细介绍之前，我们先来了解一些聚类分析的例子。

顾客偏爱：设想你运营一个大型的在线商城，想为用户提供个性化购物体验。你的目的并非完全利他的——而是希望通过改善购物体验，增加用户消费。一种方法是向每一位用户提供一套独特的建议，即让他们看到访问你网站的时间。虽然你不能直接知道每个用户的个人喜好和品味，但你有大量的数据——每个用户产生的全部购买记录。这是经典的机器学习问题——没有基本模型，而只有大量数据。

假设我们能够基于消费者的购物历史定义他们之间的一种相似性度量，就可以利用聚类分析将消费者分为 K 组。在每组中，消费者具有相似的购物模式。同一组中消费者的差异，可以作为推荐系统的基础。例如，因为消费某些相同的商品，消费者 A 和 B 同属于一个聚类——也许他们都对某种体育项目感兴趣。然而，消费者 A 又另外买了一些商品，而 B 没有。根据相似性的大小，为消费者 B 推荐这些商品是有一些道理的。

也可以利用购买的消费者信息，通过聚类分析商品来建立推荐系统。如果商品 1 和 2 都被消费者 A、D、F 和 C 购买，则可以认为这些商品是相似的。然后可以为消费者推荐与其已经购买商品相似（在这个意义上说）的商品。

基因功能预测：分子生物学中大量研究工作涉及将基因分类到特定功能类别的问题——一个具体基因发挥什么作用？它的目的是什么？一个潜在的信息来源是 mRNA 基因芯片数据（microarray data）——描述基因在特定生物样本中活动的数值量。随着时间的推移，对于一组基因，这种活动是可以测量到的。如果基于这种表示方法聚类基因，则得到一个基因的分组，使得同一组内基因随着时间推移表现出相似的行为。考虑一个包括 10 个基因的组（聚类），已知其中一半的基因功能，而另一半的功能是未知的。在没有其他证据的情况下，假设未知的一半基因具有与功能已知基因相同或相似功能也许是合理的。这不会总是得到正确的功能，但对于进一步的分析是一个很好的开始。

在这个例子中，聚类分析所得到的数据结构使我们能够做出某些与对象有关的预测。有趣的是，这个问题也可以看做是有监督分类问题，其中已知功能的基因作为训练集（类别标

签由不同的功能组成)、未知的基因作为测试集由算法对其进行类别标记。

6.2 K 均值聚类

图 6-1 所示的数据中, 包括 100 个对象 x_1, \dots, x_{100} , 每个对象由两个属性表示: $\mathbf{x} = [x_1, x_2]^T$ 。在我们绘制分类数据图时, 属于不同类别的对象用不同的符号表示。现在我们没有类别信息——所有的点看起来是一样的。

如果手动将这些对象分为包含相似对象的组时, 你可能会得到这样的结论, 有 3 组。虽然有少部分比较难分 (如点 $\mathbf{x} \approx [2.5, -1]^T$), 但大部分对象很容易分入 3 个组中之一。

通过这种方式聚类分析数据, 我们隐式地定义了相似性的含义——相似的对象是指相互之间距离平方近的对象 (如果 $(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2$ 较小, 则 i 和 j 相似)。在没有与数据或聚类分析目标相关的附加信息情况下, 这是一个合理的相似性度量。还有一些可能更合适的其他相似性定义方法, 例如马氏 (Mahalanobis) 距离 $(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j)$ 。这些距离都适合实数数据。而对于其他类型 (如文本) 的数据, 则需要不同的距离度量。

为了设计能够自动实现这种分组的算法, 我们需要更形式化地定义什么是聚类。K 均值将聚类定义为具有代表性的点, 就像一个数据对象。该点为聚类中对象的均值 (mean) (因而称为 K 均值)。我们将用 μ_k 表示第 k 个聚类的平均点; z_{nk} 为一个二值标志变量, 其中 1 表示对象 n 被分配到聚类 k 中; 否则, 为 0。每个对象必须并且只能分配到一个聚类中, 即 $\sum_k z_{nk} = 1$ 。由此我们得出如下 μ_k 的表达式:

$$\mu_k = \frac{\sum_n z_{nk} \mathbf{x}_n}{\sum_n z_{nk}} \quad (6-1)$$

每个对象被分配到最近的聚类, 即使 $(\mathbf{x}_n - \mu_k)^T (\mathbf{x}_n - \mu_k)$ (或者其他适合的距离) 的值最小的聚类 k 。

这是一个循环的推理: 将分配到这些点上的中心定义聚类, 同时点又被分配到它们最近的聚类。如果知道聚类 μ_1, \dots, μ_K , 我们就可以计算这些点的分配, 但如果没有这些分配, 就无法计算聚类。K 均值聚类通过一种迭代方案解决了这个问题。从聚类均值 μ_1, \dots, μ_K 的初始 (随机) 值开始:

- 1) 对每个数据对象 \mathbf{x}_n , 找到使 $(\mathbf{x}_n - \mu_k)^T (\mathbf{x}_n - \mu_k)$ (即找到距离最近的聚类均值) 最小的 k , 并设置 $z_{nk} = 1$ 和 $z_{nj} = 0$, 满足所有的 $j \neq k$ 。
- 2) 如果所有分配 (z_{nk}) 较前一次迭代没有变化, 则停止。
- 3) 按照式 (6-1) 更新每个 μ_k 。
- 4) 返回到 1)。

图 6-2 描述了该算法对图 6-1 所表示数据的执行过程 (MATLAB 脚本: kmeansexample.m)。图 6-2a 表示的是对均值的初始选择 (较大的符号) 和带有与最近均值相同符号的数据对象。利用式 (6-1) 更新均值, 图 6-2b 显示了这些均值向它们的新位置移动。现在,

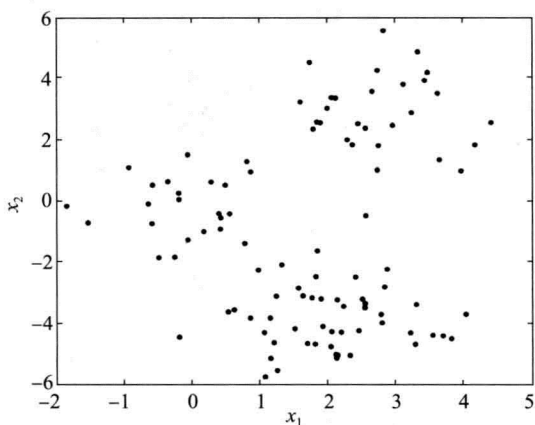


图 6-1 聚类实例的仿真数据

均值已经改变，需要重新分配对象，图 6-2c 表示了新的分配和均值结果的变化。图 6-2d、e 分别表示 3 次和 5 次迭代后的状态。在 8 次迭代后，算法已经收敛，最终的分配（聚类）结果如图 6-2f 所示。点在 $x_n = [2.5, -1]^T$ 似乎被错误地分配——这是由于坐标尺度的问题。

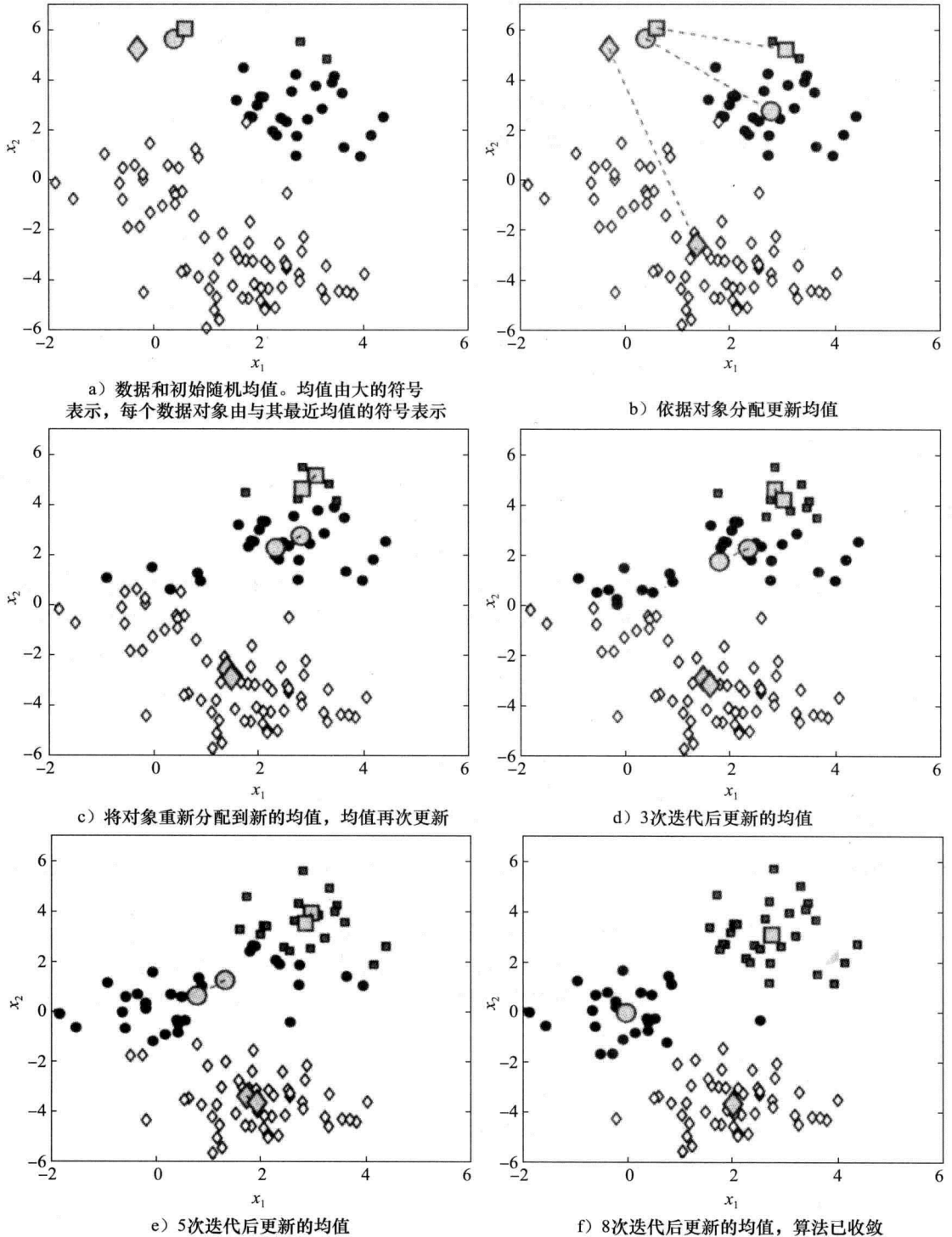


图 6-2 K 均值算法描述。数据对象由小的符号表示，均值由大的符号表示。对象由其被分配均值的符号表示

这种迭代方案能够保证收敛到下面值的局部最小：

$$D = \sum_{n=1}^N \sum_{k=1}^K z_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T (\mathbf{x}_n - \boldsymbol{\mu}_k) \quad (6-2)$$

式(6-2)可以理解为所有对象与它们对应聚类中心的距离之和。然而,并不能保证得到可能的最小值(即全局最小值)。是否能够得到全局最小值取决于聚类均值的初始选择。对与 K 均值算法,该问题是无法完全解决的,除非我们能够评价 N 个对象与 K 个聚类的所有分配方式,而这即使对于很小的 N 和 K 都是不可行的。一定程度克服这种不足的更常用方法是从多个随机初始点运行算法并选择总距离最小的解。

6.2.1 聚类数目的选择

为了使用 K 均值,我们需要选择 K 值——聚类数目。确定聚类的数目是聚类分析中常见的问题。前面讲到, K 均值产生与式(6-2)的局部最小值相对应的聚类结果。遗憾的是, D 并不完全合适,这很类似于似然是一个差的模型选择标准(它单调递增,而模型更加复杂,如图2-11a所示)。图6-3显示了当 K 增加时的 $\log D$ (MATLAB脚本:kmeanK.m)。对每个 K 值,我们采用50次算法的随机初始化,盒状图表示中间值、25%、75%以及离群点。很明显, $\log D$ (也就是 D)随着 K 的增加而降低。当 K 增加时,较大的聚类将被分解为更小的部分。聚类越小,每个点离它的聚类均值越近(平均),减少了其对 D 值的贡献。考虑 $K=N$ 的极端情况,当每个聚类只包含一个对象同时 $\mathbf{u}_k = \mathbf{x}_n$ 时, $D=0$ 。

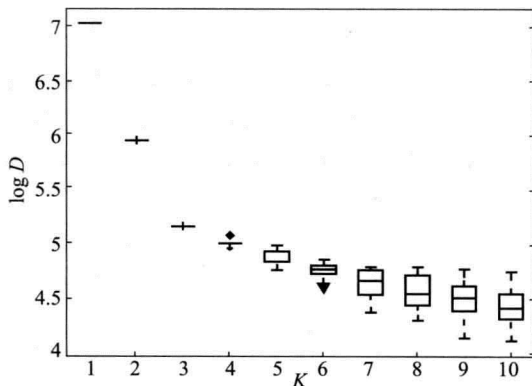


图6-3 在图6-1所示数据上,当 K 增加时的 $\log D$ (D 在式(6-2)中定义)。每个盒状图为 K 均值算法的50次随机初始化结果

这种模型选择问题没有简单的解决方案。为了解决该问题,将聚类分析超越于分析的总目标通常是有帮助的。例如,在6.1节,我们提到基于客户聚类的推荐系统。完成分组是为了获得数据的一个精简表示,并提供客户-产品推荐。因此,在某些验证数据上,选择能够产生最好推荐的聚类数目也许是合理的。类似地,聚类是分类中特征选择的常用方法——基于对象的值聚类特征,而非聚类对象(\mathbf{X}^T 而不是 T)。在该例中, K 应该选择使分类性能最优的值。

6.2.2 K 均值的不足之处

图6-4描述了两个 K 均值无法获取其中看似正确聚类结构的数据集。在两个例子中,真实聚类对象不一定符合我们目前相似性(距离)的定义。在第一个例子中,图6-4a,数据位于同心圆中。在这种情况下,标准的 K 均值由于两个圆的均值位置相同,显然无法正常使用。在第二个例子中,图6-4b,聚类被拉伸(注意,坐标轴的缩放)成右边聚类上方的对象与左边聚类均值更近(在图中均值为较大的符号)。

在6.2.3节中,我们将通过核化(kernelising) K 均值算法聚类图6-4a中的数据。对于图6-4b中的数据,我们将采用另一种聚类方法:混合模型(mixture models)(从6.3节开始介绍)。

6.2.3 核化 K 均值

我们使用第5章介绍的核替换方法,拓展 K 均值算法。从概念上讲,与其思想相同:

我们将数据变换到算法能够处理的空间，而不增加算法复杂性。我们利用图 6-4a 中的数据介绍这种方法。

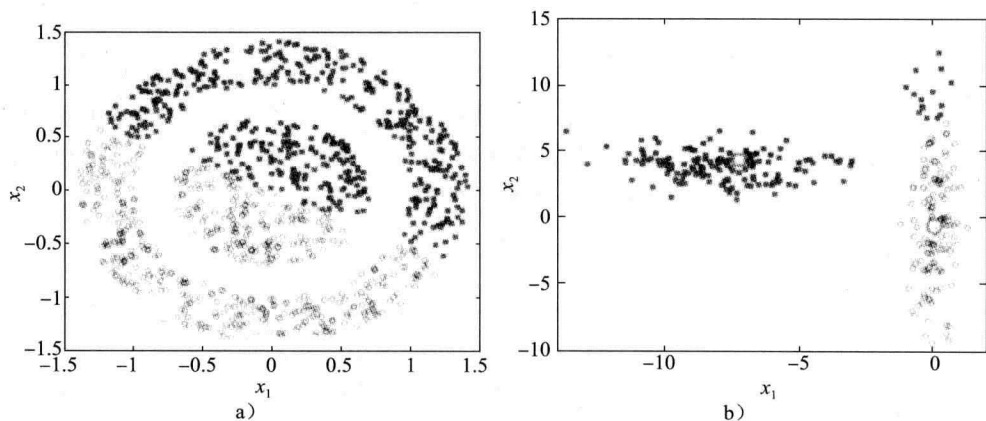


图 6-4 K 均值算法无法获取准确聚类结构的两种数据集

我们知道，核方法采用核函数直接计算变换后空间中的内（点）积，而不是对数据进行实际变换。因此，只要以内积的形式出现数据对象 $\mathbf{x}_1, \dots, \mathbf{x}_N$ ，任何算法都可以用核方法处理，使其更加强大而不额外增加任何计算成本。K 均值的关键步骤是计算第 n 个对象与第 k 个均值之间的距离：

$$d_{nk} = (\mathbf{x}_n - \boldsymbol{\mu}_k)^T (\mathbf{x}_n - \boldsymbol{\mu}_k)$$

其中，均值 $\boldsymbol{\mu}_k$ 按式 (6-1) 计算。将式 (6-1) 代入 d_{nk} 中，得到：

$$d_{nk} = \left(\mathbf{x}_n - \frac{1}{N_k} \sum_{m=1}^{N_k} z_{mk} \mathbf{x}_m \right)^T \left(\mathbf{x}_n - \frac{1}{N_k} \sum_{r=1}^{N_k} z_{rk} \mathbf{x}_r \right)$$

其中， $N_k = \sum_{n=1}^N z_{nk}$ 为分配到聚类 k 中的所有对象数目。

展开该式得出仅以数据 (\mathbf{x}_n) 的内积项表示的形式：

$$d_{nk} = \mathbf{x}_n^T \mathbf{x}_n - \frac{2}{N_k} \sum_{m=1}^{N_k} z_{mk} \mathbf{x}_n^T \mathbf{x}_m + \frac{1}{N_k^2} \sum_{m=1}^{N_k} \sum_{r=1}^{N_k} z_{mk} z_{rk} \mathbf{x}_m^T \mathbf{x}_r$$

将所有内积替换为核函数，得到核化距离：

$$d_{nk} = K(\mathbf{x}_n, \mathbf{x}_n) - \frac{2}{N_k} \sum_{m=1}^{N_k} z_{mk} K(\mathbf{x}_n, \mathbf{x}_m) + \frac{1}{N_k^2} \sum_{m=1}^{N_k} \sum_{r=1}^{N_k} z_{mk} z_{rk} K(\mathbf{x}_m, \mathbf{x}_r) \quad (6-3)$$

该距离是完全关于数据和当前分配的函数，没有出现聚类均值。事实上，在变换后的空间中计算聚类均值一般不太可能。聚类 K 均值的原始表达为：

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N z_{nk} \mathbf{x}_n}{\sum_{n=1}^N z_{nk}}$$

其核化后的版本为：

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N z_{nk} \phi(\mathbf{x}_n)}{\sum_{n=1}^N z_{nk}}$$

在该式中，数据对象以其自身形式出现，而不是内积形式。在第 5 章中对于大多数核函数，我们讨论了无法计算变换 $(\mathbf{x}_n \rightarrow \phi(\mathbf{x}_n))$ 的情况：我们只计算变换后空间中的内积

$(\phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m))$ 。如果不能计算该变换，我们将无法计算 μ_k 。

式 (6-3) 给出核化 K 均值步骤：

- 1) 对每个 n 随机初始化 z_{nk} 。
- 2) 利用式 (6-3) 计算每个对象的 d_{n1}, \dots, d_{nK} 。
- 3) 将每个对象分配到 d_{nk} 最近的聚类中。
- 4) 如果分配有变化，则返回到 2)；否则，停止。

在标准的 K 均值中，我们用随机设置均值 μ_1, \dots, μ_K 的方法初始化算法。在核 K 均值中，由于不能获得均值，所以通过对对象-聚类的分配初始化算法。我们知道 K 均值算法对初始条件敏感，我们可以完全随机地初始化——对每个 n 设置 z_{nk} 为 1，所有其他 ($z_{nl}, l \neq k$) 为 0，但更加小心会更好。不同的方法是，运行标准 K 均值并使用收敛时的 z_{nk} 值。其优点是能够使得在同一聚类中的对象与其他对象具有合理的距离（这是随机设置无法保证的）。第二种方法是，将 $N-K+1$ 对象分配到聚类 1 中，将剩下的 $K-1$ 个对象分别分配到其余聚类中。每次迭代的性能依赖于被聚类数据的具体特点。

图 6-5 给出了用核 K 均值处理图 6-4a 中数据的结构 (MATLAB 脚本: kernelkmeans.m)。既然如此，我们将除一个对象之外的所有对象分配到“圆形”聚类中，余下的对象分配到“方形”聚类中。我们采用 $\gamma=1$ 的高斯核 (式 (5-19))。图 6-5a 显示了初始化后一次迭代的分配情况。在算法经过 5、10 和 30 次迭代 (分别为图 6-5b、c 和 d) 后，较小的聚类占据了内部的圆。收敛后，可以看到算法得到了数据中有意义的结构。

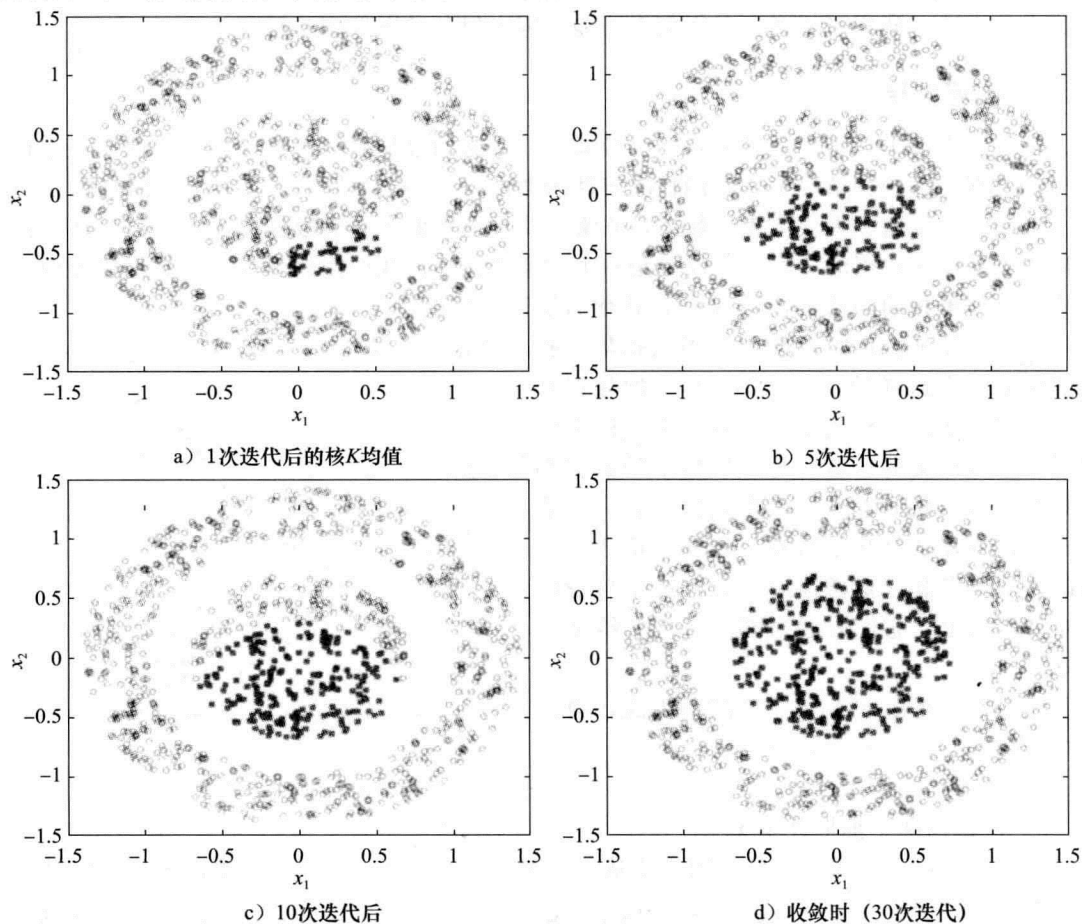


图 6-5 核化 K 均值在图 6-4a 所示数据上的执行结果

在聚类时，核 K 均值方法不仅没有改变最初的相似性思想，而且还为分析其他数据类型提供了方法。我们能够处理任何存在核函数的数据类型，而且几乎没有数据类型不存在核函数。明显的例子是用于文本的核（每个对象是一个文档）以及图或网络的核，后者在计算生物学中广泛使用。

6.2.4 小结

在前面的各节中，我们介绍了 K 均值算法以及如何将其进行核化。 K 均值算法的一个突出优点是它容易使用，并且没有很大的计算挑战。然而，其简单性也是一个缺点：假设聚类可以表示为单个点（均值）往往会过于粗糙。另外，如果我们的目标仅仅是聚类（我们曾经提到过如何选择聚类数目，如在分类任务中得到最好的性能等），那么没有指导方法用来决定的聚类数目。为了解决这些缺点，我们现在介绍利用统计混合模型进行聚类的方法。这些模型与 K 均值有一些相似，但提供了更丰富的数据表达形式。

6.3 混合模型

在图 6-4b 中，我们展示了一个原始 K 均值聚类失败的数据集。本应该属于一个类的某些对象却更靠近另一个类的中心，两个类以这样的方式被拉大了。我们这里的 K 均值算法的问题是关于类的定义过于粗糙。这些延伸类的特性不能由一个单点和平方距离来描述。我们需要能够纳入形状的概念。统计混合学将每个类表示为一个概率密度。这种归纳引出了一个强大的方法，我们可以在几乎任何类型的数据中以各种图形来建模聚类。

6.3.1 生成过程

在 2.1.1 节中，我们通过创建一个能够生成数据的过程对第 1 章介绍的线性模型进行了概率化处理。在这种情况下，我们将决定性函数 $w^T x_n$ 与均值为 0、方差为 σ^2 的高斯随机变量结合起来。以这种方式产生的数据，非常高质量地类似于真实数据。注意，我们从未试图声称这就是产生数据的过程，它仅仅是一个抽象概念，使我们能够建立一个更好的模型。我们将使用很多相同的动机，从 K 均值转化为统计混合模型。

图 6-6 再现了我们合成的聚类数据集。我们如何能产生这样的数据？图 6-6 中的数据并不像我们所遇到的任何密度函数样本。图中出现了 3 个不相交的区域，每个区域中的数据都聚集在一起。我们看到的密度函数没有一个能够产生这样复杂结构的数据。然而，这 3 个区域的每一个看起来足够简单，能够产生自身的分布。事实上，它们看起来都有点像 2 维高斯样本。

假设数据是由 3 个独立的高斯生成，通过两个步骤来抽样第 n 个数据对象 x_n ：

- 1) 从 3 个高斯中选择一个。
- 2) 从该高斯中抽取样本 x_n 。

这些步骤都是简单明确的。第 1 步从一个离散集中选择一个值，像滚动骰子。要做到这一点，我们只需要定义每个输出 π_k 的概率，满足 $\sum_k \pi_k = 1$ 。选择了从哪个高斯进行抽样，

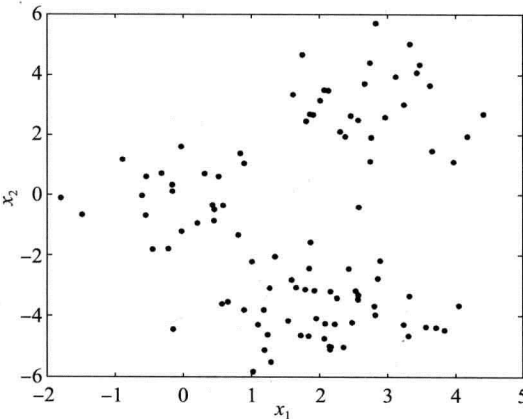


图 6-6 合成的聚类数据集的例子

214
}
245

216

第二步就很简单了。

为了阐明这个过程, 我们将从一个设置 $k=2$ 的高斯分布中取样数据。在 K 均值中, 我们使用 z_{nk} 作为指示变量。如果我们选择第 k 个组分 (component) 作为第 n 个对象的来源, 那么我们设置 $z_{nk}=1$, 并且对其他的 $j \neq k$, 设置 $z_{nj}=0$ 。 μ_k 和 Σ_k 表示第 k 个高斯分布的参数。

如果 x_n 是从第 k 个组分中产生的, 那么它的密度函数为一个均值和协方差分别为 μ_k 和 Σ_k 的高斯分布:

$$p(x_n | z_{nk} = 1, \mu_k, \Sigma_k) = \mathcal{N}(\mu_k, \Sigma_k)$$

在我们的例子中, 对这 2 个组分我们采用如下的均值和协方差,

$$\mu_1 = [3, 3]^T, \quad \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad \mu_2 = [1, -3]^T, \quad \Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \quad (6-4)$$

最后, 我们需要定义 π_k 。如果第一个组分比第二个组分更可能, 那么我们使用 $\pi_1=0.7$ 、 $\pi_2=0.3$ 。图 6-7 展示了生成的前 50 个数据和这两个高斯分布的密度函数 (MATLAB 脚本: mix-gen.m)。我们选择 $k=2$, 图 6-7a 描述了选出的第 1 个点, 它是从第 2 个组分中选出来的 (比较低的那个组分)。图 6-7b 描述了选出的前 5 个点, 注意这 5 个点中除了第 1 个点外都来自于第 1 个组分。这不奇怪, 因为第 1 个组分比第 2 个组分更可能 $\pi_1 > \pi_2$ 。图 6-7c 和 d 分别描述了前 10 个点和前 50 个点。如果我们比较图 6-7d 和图 6-6, 我们就会发现尽管数据集不一样, 但是它们有一些共同的特点。特别地, 图 6-6 看起来就像与图 6-7 用相同的方式生成的。

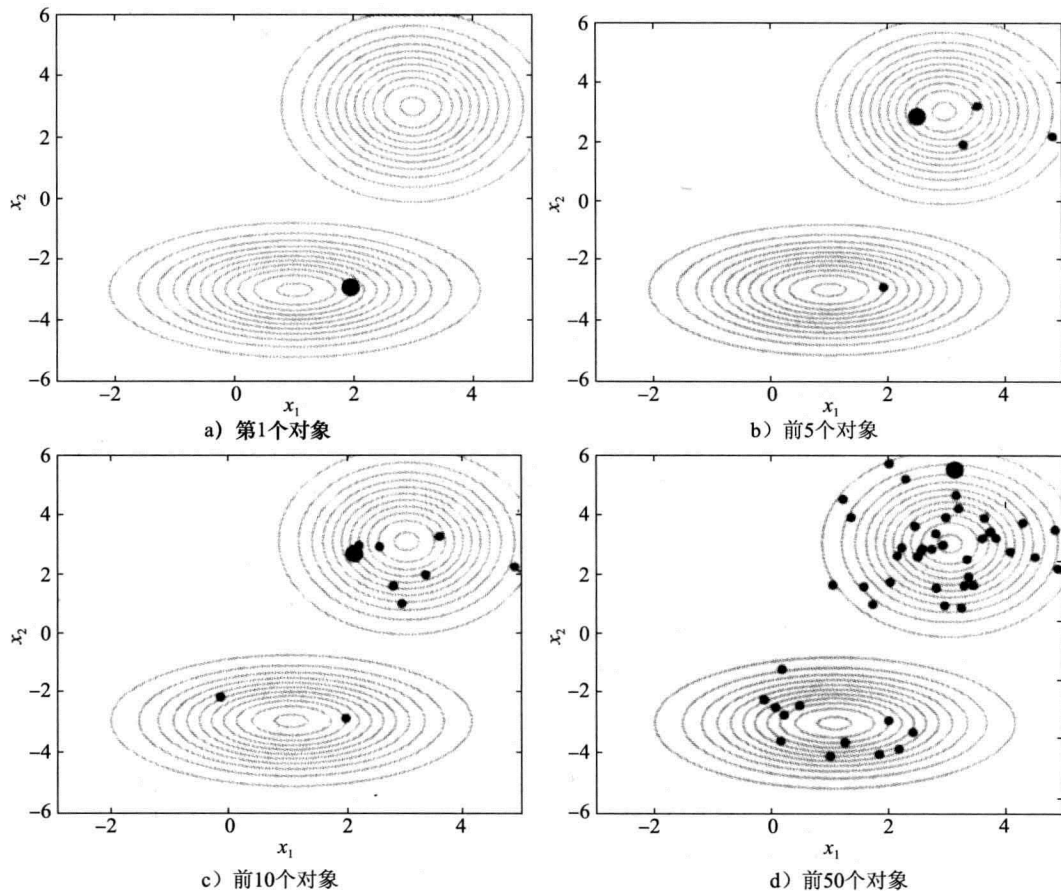


图 6-7 从两个高斯分布中生成数据

以上我们介绍的生成程序是针对混合模型（假定数据从一些不同密度函数构成的混合函数中抽取）的生成程序。因为拟合一些简单的分布往往比拟合一个复杂的分布更简单，所以混合模型在数据模型中有着广泛的应用。在聚类分析中，我们可以把每一个单独的组分作为一个类—— $z_{nk}=1$ 的所有个体都在第 k 个类中。我们的学习任务是从已知的数据中，推断出每个组分的参数 (μ_k, Σ_k) 和各个组分中个体的分配。在 K 均值中，这是一个循环问题：如果我们知道个体的分配，那么计算参数非常容易；同样，如果知道分布的参数，那么个体的分配也非常容易。如果两个都不知道，那么我们很难知道从何开始。期望-最大 (EM) 算法给我们一个结果，EM 算法是一个在很多模型中广泛应用的迭代最大似然技术，并且它与我们之前介绍的 K 均值算法是并行的算法。

6.3.2 混合模型似然函数

为了进行 EM 算法，我们需要首先给出似然函数的表达式。为了尽可能地通用化，我们用 $p(\mathbf{x}_n | z_{nk}=1, \Delta_k)$ 表示第 k 个类的密度函数（不一定为高斯分布），其中 Δ_k 为其中的参数。另外，我们用 $\Delta = \{\Delta_1, \dots, \Delta_K\}$ 来表示各个组分的参数集合，并把所有的 π_k 整合为一个向量 $\pi = \{\pi_1, \dots, \pi_K\}$ 。

我们需要在整个模型下数据 \mathbf{x}_n 的似然函数 $p(\mathbf{x}_n | \Delta, \pi)$ 。为了得到这个表达式，我们从 $z_{nk}=1$ 的特定数据对象的似然函数开始：

$$p(\mathbf{x}_n | z_{nk}=1, \Delta) = p(\mathbf{x}_n | \Delta_k)$$

为了得到 $p(\mathbf{x}_n | \Delta, \pi)$ ，我们需要删除 z_{nk} 。为了实现它，我们首先在等式两边都乘以 $p(z_{nk}=1)$ ，也就是我们之前定义的 π_k 。那么有

$$p(\mathbf{x}_n | z_{nk}=1, \Delta) p(z_{nk}=1) = p(\mathbf{x}_n | \Delta_k) p(z_{nk}=1)$$

$$p(\mathbf{x}_n, z_{nk}=1 | \Delta, \pi) = p(\mathbf{x}_n | \Delta_k) \pi_k$$

等式两边对所有的 k 个组分进行求和，得到似然函数

$$\begin{aligned} \sum_{k=1}^K p(\mathbf{x}_n, z_{nk}=1 | \Delta, \pi) &= \sum_{k=1}^K p(\mathbf{x}_n | \Delta_k) \pi_k \\ p(\mathbf{x}_n | \Delta, \pi) &= \sum_{k=1}^K \pi_k p(\mathbf{x}_n | \Delta_k) \end{aligned}$$

根据基本的样本独立性假设，我们可以得到 N 个数据对象的似然函数：

$$p(\mathbf{X} | \Delta, \pi) = \prod_{n=1}^N \sum_{k=1}^K \pi_k p(\mathbf{x}_n | \Delta_k) \quad (6-5)$$

6.3.3 EM 算法

我们现在需要说明使用 EM 算法求式 (6-5) 中似然函数的最大值。通常使用对数似然函数更为简单，因此我们对式 (6-5) 取自然对数，即，

$$L = \log p(\mathbf{X} | \Delta, \pi) = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k p(\mathbf{x}_n | \mu_k, \Sigma_k) \quad (6-6)$$

对数内的求和使我们直接寻找最优的 μ_k 、 Σ_k 、 π 参数值比较困难。而 EM 算法通过计算似然函数的一个下界 (\mathbf{X} 、 Δ 和 π 的一个函数且总是小于等于似然函数 L) 来解决这个问题。我们不再直接对 L 进行最大化，而转为对它的下界最大化。

为了得到 L 的下界，我们可以使用下面期望的对数和期望的对数的期望关系，也就是著名的詹森 (Jensen) 不等式：

$$\log \mathbf{E}_{p(z)} \{f(z)\} \geq \mathbf{E}_{p(z)} \{\log f(z)\} \quad (6-7)$$

也就是说, $f(z)$ 期望值的对数总是大于等于 $\log f(z)$ 的期望值。

为了能够应用詹森不等式来求似然函数的下界, 我们需要式 (6-6) 的右侧部分看起来像期望的对数。因此, 将对 k 求和的公式内的表达式先乘后除以一个新的变量 q_{nk} 。

$$L = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \frac{q_{nk}}{q_{nk}}$$

如果我们约束 q_{nk} 是正的且满足求和约束条件 $\sum_{k=1}^K q_{nk} = 1$ (也就是说, q_{nk} 表示第 n 个个体在这个 k 个组分中的概率分布), 那么我们可以重新整理公式为基于 q_{nk} 的期望, 即,

$$\begin{aligned} L &= \sum_{n=1}^N \log \sum_{k=1}^K q_{nk} \frac{\pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{q_{nk}} \\ &= \sum_{n=1}^N \log \mathbf{E}_{q_{nk}} \left\{ \frac{\pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{q_{nk}} \right\} \end{aligned}$$

利用詹森不等式, 我们可以得到 L 的下界,

$$L = \sum_{n=1}^N \log \mathbf{E}_{q_{nk}} \left\{ \frac{\pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{q_{nk}} \right\} \geq \sum_{n=1}^N \mathbf{E}_{q_{nk}} \left\{ \log \frac{\pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{q_{nk}} \right\}$$

不等式的右侧部分就是我们需要优化的表达式的下界 (记为 \mathcal{B})。把表达式展开, 我们将更容易操作。

$$\begin{aligned} \mathcal{B} &= \sum_{n=1}^N \mathbf{E}_{q_{nk}} \left\{ \log \frac{\pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{q_{nk}} \right\} \\ &= \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log \left(\frac{\pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{q_{nk}} \right) \\ &= \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log \pi_k + \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) - \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log q_{nk} \end{aligned} \quad (6-8)$$

使得这个下界达到局部最大值的 q_{nk} 、 π_k 、 $\boldsymbol{\mu}_k$ 、 $\boldsymbol{\Sigma}_k$ 参数值也会使对数似然函数 L 达到最大。

就像我们前面提到的, EM 算法是一个迭代算法。这就需要我们不断地重复更新模型中的数值直到收敛。为了每次更新, 我们需要计算 \mathcal{B} 针对某个参数的偏导数, 并令其等于 0, 然后求解。下面我们将对各个参数依次求解。

6.3.3.1 更新 π_k

只有 \mathcal{B} 的第一部分包含 π_k (其他部分对 π_k 的偏导数为 0)。 π_k 是一个概率, 所以有 $\sum_k \pi_k = 1$ 。

因此, 对 π_k 进行优化是有条件约束的。就像在 5.3.3.2 节的 SVM 中一样, 可以采用拉格朗日算法将约束条件整合进目标函数 (此时为 \mathcal{B})。与 \mathcal{B} 相关的拉格朗日项 (和关联的拉格朗日乘子 λ , 见注解 5.1) 为,

$$\mathcal{B} = \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log \pi_k - \lambda \left(\sum_{k=1}^K \pi_k - 1 \right) + \dots$$

对上式求 π_k 的偏导数, 并使其等于 0, 然后整理, 得到,

$$\begin{aligned} \frac{\partial \mathcal{B}}{\partial \pi_k} &= \frac{\sum_{n=1}^N q_{nk}}{\pi_k} - \lambda = 0 \\ \sum_{n=1}^N q_{nk} &= \lambda \pi_k \end{aligned} \quad (6-9)$$

最后我们需要计算 λ , 式 (6-9) 两边对 k 求和, 可得:

219

220

$$\begin{aligned}\sum_{k=1}^K \sum_{n=1}^N q_{nk} &= \lambda \sum_{k=1}^K \pi_k \\ \sum_{n=1}^N 1 &= \lambda \\ \lambda &= N\end{aligned}$$

其中我们用到了 $\sum_{k=1}^K q_{nk} = 1$ 和 $\sum_{k=1}^K \pi_k = 1$ 的事实。将 $\lambda = N$ 代入式 (6-9)，可以得到 π_k 的表达式为：

$$\pi_k = \frac{1}{N} \sum_{n=1}^N q_{nk}$$

我们将在 6.3.3.5 节讨论它和其他表达式的直觉意义。

6.3.3.2 更新 μ_k

接下来，我们考虑 μ_k ， \mathcal{B} 中只有第 2 部分包含 μ_k 。如果我们将 $p(\mathbf{x}_n | \mu_k, \Sigma_k)$ 作为多变量高斯分布的密度函数（式 (2-28)），并展开，可得：

$$\begin{aligned}\mathcal{B} &= \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log \left(\frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{x}_n - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_n - \mu_k) \right) \right) + \dots \\ &= -\frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log((2\pi)^d |\Sigma_k|) - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K q_{nk} (\mathbf{x}_n - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_n - \mu_k) + \dots\end{aligned}$$

第一部分不包含 μ_k ，因此可以忽略。利用下面的性质（见表 1-4），

$$f(\mathbf{w}) = \mathbf{w}^T \mathbf{C} \mathbf{w}, \quad \frac{\partial f(\mathbf{w})}{\partial \mathbf{w}} = 2\mathbf{C} \mathbf{w}$$

和链式法则，我们可以求 \mathcal{B} 对 μ_k 的偏导数，

$$\begin{aligned}\frac{\partial \mathcal{B}}{\partial \mu_k} &= -\frac{1}{2} \sum_{n=1}^N q_{nk} \times \frac{\partial (\mathbf{x}_n - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_n - \mu_k)}{\partial (\mathbf{x}_n - \mu_k)} \times \frac{\partial (\mathbf{x}_n - \mu_k)}{\partial \mu_k} \\ &= \sum_{n=1}^N q_{nk} \Sigma_k^{-1} (\mathbf{x}_n - \mu_k)\end{aligned}$$

令其等于 0 并整理，可以得到 μ_k 的表达式，

$$\begin{aligned}\sum_{n=1}^N q_{nk} \Sigma_k^{-1} (\mathbf{x}_n - \mu_k) &= 0 \\ \sum_{n=1}^N q_{nk} \Sigma_k^{-1} \mathbf{x}_n &= \sum_{n=1}^N q_{nk} \Sigma_k^{-1} \mu_k \\ \sum_{n=1}^N q_{nk} \mathbf{x}_n &= \mu_k \sum_{n=1}^N q_{nk} \\ \mu_k &= \frac{\sum_{n=1}^N q_{nk} \mathbf{x}_n}{\sum_{n=1}^N q_{nk}}\end{aligned} \quad (6-10)$$

6.3.3.3 更新 Σ_k

第三，我们考虑 Σ_k 。与 μ_k 一样，我们只需要考虑 \mathcal{B} 中的项 $p(\mathbf{x}_n | \mu_k, \Sigma_k)$ 。我们将该项展开

$$\mathcal{B} = -\frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log((2\pi)^d |\Sigma_k|) - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K q_{nk} (\mathbf{x}_n - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_n - \mu_k) + \dots$$

忽略第一项中的常数部分 (2π) ，我们得到，

$$\mathcal{B} = -\frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log(|\Sigma_k|) - \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K q_{nk} (\mathbf{x}_n - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_n - \mu_k) + \dots$$

为了对 Σ_k 求偏导数, 我们需要以下的性质:

$$\frac{\partial \log |\mathbf{C}|}{\partial \mathbf{C}} = (\mathbf{C}^T)^{-1}$$

和

$$\frac{\partial \mathbf{a}^T \mathbf{C}^{-1} \mathbf{b}}{\partial \mathbf{C}} = -(\mathbf{C}^T)^{-1} \mathbf{a} \mathbf{b}^T (\mathbf{C}^T)^{-1}$$

利用这两个性质, 我们可以求 \mathcal{B} 对 Σ_k 的偏导数。

$$\frac{\partial \mathcal{B}}{\partial \Sigma_k} = -\frac{1}{2} \sum_{n=1}^N q_{nk} \Sigma_k^{-1} + \frac{1}{2} \sum_{n=1}^N q_{nk} \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}$$

注意, Σ_k 是一个协方差矩阵, 是对称的, 因此 $\Sigma_k^T = \Sigma_k$ 。令该式等于 0 并整理, 得到

$$\begin{aligned} -\frac{1}{2} \sum_{n=1}^N q_{nk} \Sigma_k^{-1} + \frac{1}{2} \sum_{n=1}^N q_{nk} \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} &= 0 \\ \frac{1}{2} \sum_{n=1}^N q_{nk} \Sigma_k^{-1} &= \frac{1}{2} \sum_{n=1}^N q_{nk} \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} \end{aligned}$$

在等式两侧分别左乘和右乘 Σ_k , 可以使我们消掉 Σ_k^{-1} :

$$\begin{aligned} \Sigma_k \sum_{n=1}^N q_{nk} \Sigma_k^{-1} \Sigma_k &= \Sigma_k \Sigma_k^{-1} \sum_{n=1}^N q_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} \Sigma_k \\ \Sigma_k \sum_{n=1}^N q_{nk} &= \sum_{n=1}^N q_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \\ \Sigma_k &= \frac{\sum_{n=1}^N q_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^N q_{nk}} \end{aligned} \quad (6-11)$$

6.3.3.4 更新 q_{nk}

最后, 我们更新 q_{nk} , 它在 \mathcal{B} 的三项中都出现。另外, 它受条件 $\sum_{k=1}^K q_{nk} = 1$ 的约束, 因此类似于更新 π_k 。我们使用拉格朗日项 (见注解 5.1)。下界 \mathcal{B} 和拉格朗日项为:

$$\begin{aligned} \mathcal{B} &= \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log \pi_k + \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log p(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k) - \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log q_{nk} \\ &\quad - \lambda \left(\sum_{k=1}^K q_{nk} - 1 \right) \end{aligned}$$

对 q_{nk} 求偏导数, 得到,

$$\frac{\partial \mathcal{B}}{\partial q_{nk}} = \log \pi_k + \log p(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k) - (1 + \log q_{nk}) - \lambda$$

注释 6.1 (微分的乘积法则): 当需要对含有相同变量的两个函数的乘积求该变量的微分时, 我们可以使用乘积法则。例如, 如果

$$f(a) = g(a)h(a)$$

那么, 根据乘积法则, 有

$$\frac{\partial f(a)}{\partial a} = g(a) \frac{\partial h(a)}{\partial a} + \frac{\partial g(a)}{\partial a} h(a)$$

例如, 求 $a \log a$ 对 a 的微分, 则有

$$a \times \frac{1}{a} + 1 \times \log(a) = 1 + \log(a)$$

其中, 对项 $q_{nk} \log q_{nk}$ 求偏导我们用到了乘积法则 (见注解 6.1)。令其等于 0, 整理并求指数, 得到了 q_{nk} 的表达式:

$$1 + \log q_{nk} + \lambda = \log \pi_k + \log p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (6-12)$$

$$\exp(\log q_{nk} + (\lambda + 1)) = \exp(\log \pi_k + \log p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))$$

$$q_{nk} \exp(\lambda + 1) = \pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

与更新 π_k 一样, 为了得到常数项 (此时为 $\exp(\lambda + 1)$), 我们对等式两边的 k 项求和, 得到:

$$\exp(\lambda + 1) \sum_{k=1}^K q_{nk} = \sum_{k=1}^K \pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (6-13)$$

$$\exp(\lambda + 1) = \sum_{k=1}^K \pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

将式 (6-13) 代入式 (6-12), 得到 q_{nk} 的表达式:

$$q_{nk} = \frac{\pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j p(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \quad (6-14)$$

6.3.3.5 一些直觉

这 4 个更新的等式分别为

$$\pi_k = \frac{1}{N} \sum_{n=1}^N q_{nk} \quad (6-15)$$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N q_{nk} \mathbf{x}_n}{\sum_{n=1}^N q_{nk}} \quad (6-16)$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^N q_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^N q_{nk}} \quad (6-17)$$

$$q_{nk} = \frac{\pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j p(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \quad (6-18)$$

前 3 个表达式非常依赖于 q_{nk} : π_k 是针对某个 k 的 q_{nk} 的均值, $\boldsymbol{\mu}_k$ 是以 q_{nk} 为权值的数据对象的均值, $\boldsymbol{\Sigma}_k$ 是加权的协方差。那么 q_{nk} 代表什么呢? 式 (6-18) 可以给我们一些直觉。乍一看, 我们发现它像一个具有先验概率 π_k 的贝叶斯规则、似然函数 $p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ 和 k 个组分的平均值得到的标准化常数。实际上, 它可以理解为计算一个个体 n 属于类 k 的后验概率问题 (非常像式 (5-2) 给出的依据贝叶斯规则的贝叶斯分类问题)。特别地,

$$p(z_{nk} = 1 | \mathbf{x}_n, \boldsymbol{\pi}, \Delta) = \frac{p(z_{nk} = 1 | \pi_k) p(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K p(z_{nj} = 1 | \pi_j) p(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = q_{nk} \quad (6-19)$$

模型的参数 $\boldsymbol{\pi}, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_k, q_{nk}$ 的取值告诉我们第 n 个个体属于第 k 类的后验概率。鉴于此, 式 (6-15)、式 (6-16)、式 (6-17) 就非常有意义了。式 (6-15) 是所有属于第 k 类的后验概率的和, 换句话说, 个体属于第 k 类的期望比例。试想当后验概率都是 1 或者 0 的这种特殊情形。此时, π_k 正好为属于第 k 类的个体的比例, $\boldsymbol{\mu}_k$ 和 $\boldsymbol{\Sigma}_k$ 正好为数据根据其属于第 k 类的后验概率为权值的加权均值和方差——属于第 k 类的后验概率比较大的数据对第 k 类的均值和方差有更大的影响。

注意前面的讨论，我们可以把4个更新分为2个集合。第一个集合包含根据固定的分配概率 q_{nk} 来估计的模型参数 π_k 、 μ_k 和 Σ_k 的更新。第二步，根据模型参数的新值来更新分配概率 q_{nk} 。这个步骤与之前介绍的K均值算法类似。 q_{nk} 的更新类似于K均值中 z_{nk} 的更新， μ_k 、 Σ_k 、 π 的更新类似于K均值中 μ_k 的更新。关键的不同在于我们是计算类别的后验概率而不是硬分配，并考虑了各组分的协方差（尽管这是一个可选择的设计——我们可以简单地假设 $\Sigma_k = I$ ）。在式(6-16)中 q_{nk} 的替换就是式(6-1)均值的更新。

这4个等式的更新给我们展示了EM算法的一个例子。前3个参数 π_k 、 μ_k 、 Σ_k 的更新，组成M步，即在 q_{nk} 条件下最大化下界的步骤，我们称为最大(M)步。 q_{nk} 的更新称为期望(E)步，因为它实际上计算了未知 z_{nk} 分配的期望值，尽管我们没有通过这种方式求出它。我们鼓励读者探索文献中EM算法的其他应用来获得EM算法的不同的推导。

6.3.4 例子

图6-8再现了我们在本章中所使用的合成数据集，并且我们将使用它来阐述6.3.3节中的EM算法的操作(MATLAB脚本: gmix.m)。与K均值十分相似的是，我们需要指定一个期望的组分数目，本例中我们采用 $K=3$ 。与K均值不同的是，我们可以使用一个很有用的测度从数据中推断类的个数，这个方法将在下面进行阐述。

在对式(6-15)到式(6-18)进行更新之前，我们需要对某些参数进行初始化。我们随机选择3个组分的均值和协方差来进行初始化。图6-9a展示了这3个得到的高斯分布的密度函数。而且，为了能够计算式(6-18)中的 q_{nk} ，我们需要初始化 π_k ，我们在这3个分量上假设一个均匀的先验分布来得到 $\pi_k = 1/K$ 。现在我们有计算式(6-18)中 q_{nk} （期望步）所需要的初始值，然后我们要对式(6-15)、式(6-16)、式(6-17)中的 π_k 、 μ_k 、 Σ_k 进行依次更新（‘M’步）。图6-9b展示了最终结果的高斯分布。

注意，只经过一次迭代，数据中的高斯分布已经展现出聚类的结构。经过第一大步后，后面的变化会略微变慢。通过图6-9c，我们发现经过5步EM迭代，上面右侧的组分已经逐渐变得独立（完全与其他两个分离），同时另两个类也逐步分离。再经过两次迭代，从图6-9d中可以看出这两个也已经分离，从中我们可以看出，只需要经过少量的迭代我们就可以使算法收敛—— q_{nk} 和模型参数的更新没有变化。图6-9e展示了收敛的结果，此时已经可以很清晰地看出各个独立的类结构。在图6-9f中，我们可以看到下界 B 和对数似然函数 L 的进化过程。两个都应该是递增的。

一般情况下，我们不关心高斯分布本身，而是更关心对象在组分间的分配——聚类。这些信息由 q_{nk} 提供——对象属于某个组分的后验概率。如果我们想把每个对象只分配到一个组分，那么我们可以把每个对象分配到后验概率最高的组分。有必要指出，类似于这样的硬分配有可能并不是最明智的。考虑一个对象 n ，它在收敛时有如下的 q_{nk} 值：

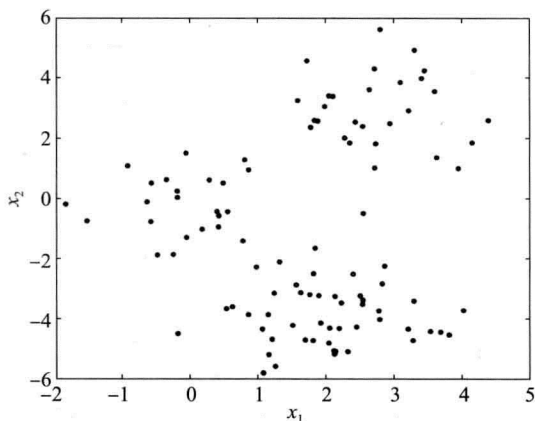
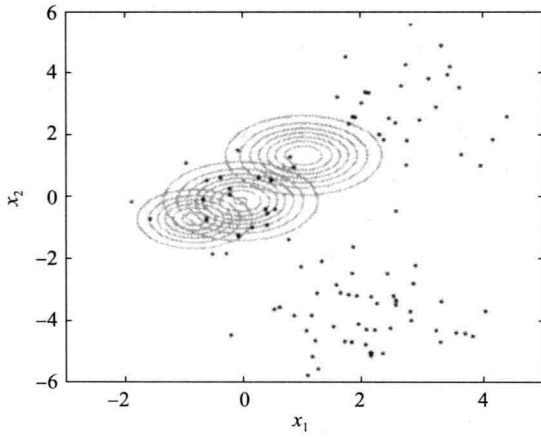
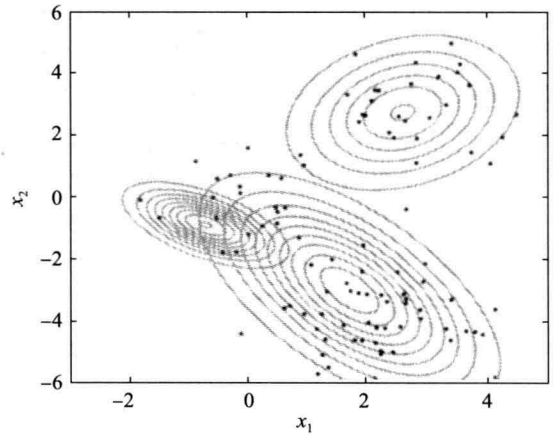


图6-8 本章中遇到的合成聚类数据

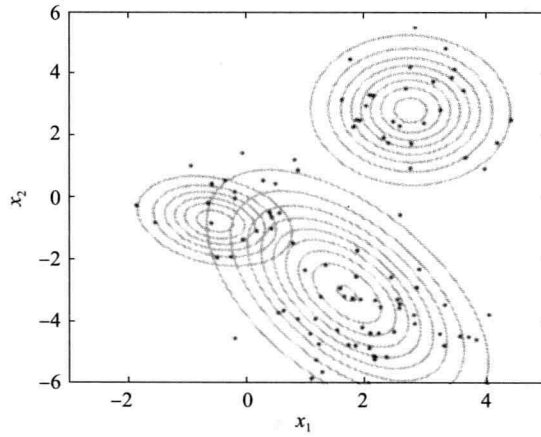
$$q_{n1} = 0.53, \quad q_{n2} = 0.45, \quad q_{n3} = 0.02$$



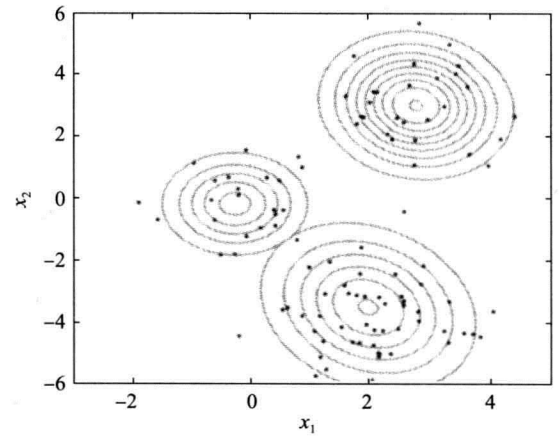
a) 随机初始化的3个高斯混合组分



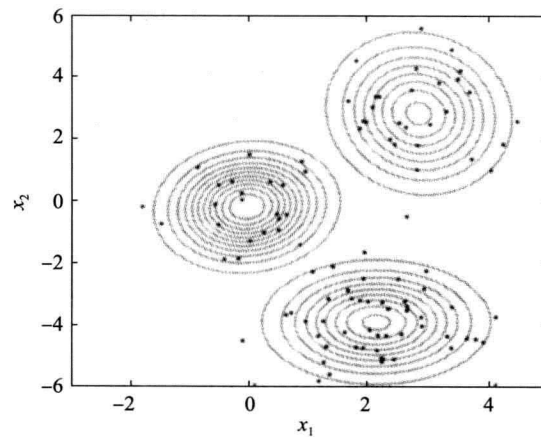
b) EM算法1次迭代后3个组分



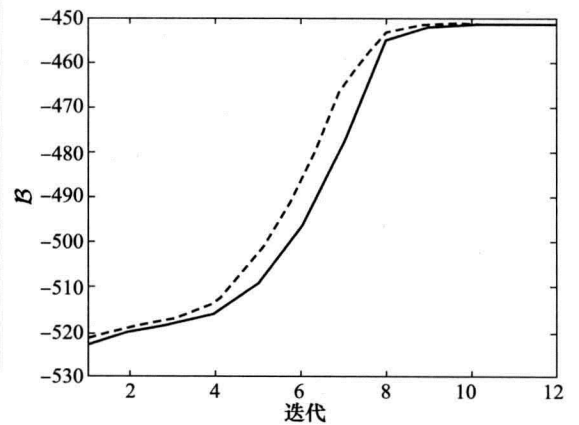
c) EM算法5次迭代后3个组分



d) EM算法7次迭代后3个组分



e) EM算法收敛后的3个组分



f) 边界B的变化(实线, 式(6-8))
和对数似然L(虚线, 式(6-5))

图 6-9 高斯混合模型的运算实例

如果我们必须将它分配到一个特定的组分，那么组分 1 是最适合的，然而如果我们这么做，我们就丢失了对对象 n 与组分 2 关系的有用信息。

从这个观点出发，你可能想知道为什么我们要通过这样相当复杂的方式来做这件事？ K 均值似乎在用一个更简单的方式来实现—— K 均值和混合模型的聚类结果是几乎相同的，并且 K 均值可以进行核化。在接下来的两节中，我们将看到混合模型相对 K 均值聚类已有一些非常重要的优势，主要是因为它们的统计学性质。

在我们进行下一步之前，我们重新审视一个数据，来展示从 K 均值（图 6-10a）到混合模型的提高。选取 $k=2$ ，并用本节的方法进行公式的更新，我们可以将混合模型应用到这个数据，图 6-10b 显示了混合模型的结果。明显地，混合模型可以很好地提取到有意义的聚类结果信息。

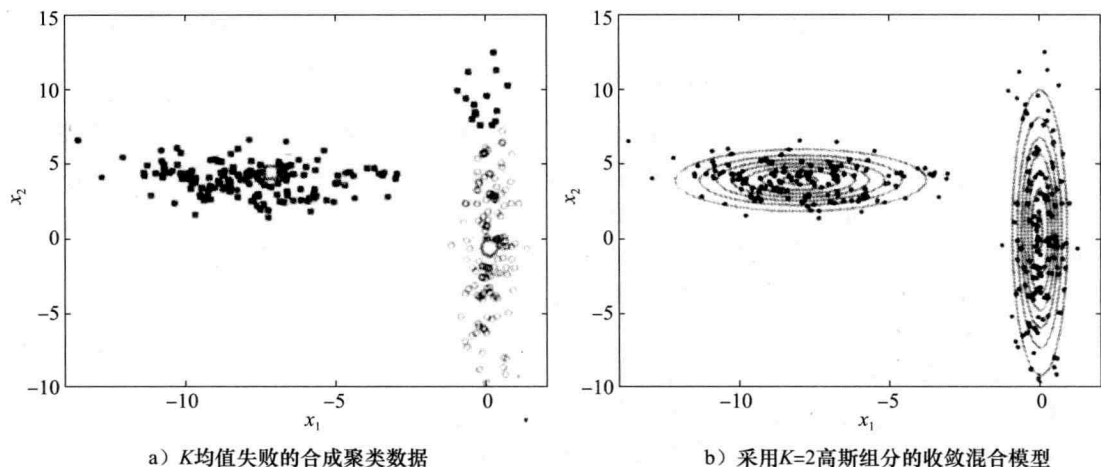


图 6-10 K 均值失败但混合模型能够解决的例子

6.3.5 EM 寻找局部最优

与 K 均值一样，EM 算法得到的聚类结果依赖于特定的初始值。它只能保证似然函数达到局部最大，而不一定是全局最大。实际上，即使我们采取同一个似然函数，如果重新定义组分标签，我们经常得到多个全局最大值。关于 K 均值，我们不能通过解析的方法解决此问题，只能凭借从多个点出发，多次运行程序来解决。我们可以通过似然函数（式（6-5））来评价哪个收敛结果更好（就像我们在 K 均值中采用式（6-2）一样）。

6.3.6 组分数目的选择

与 K 均值一样，我们必须通过选择组分的数目来指定类的个数。我们发现在 K 均值中，这个选择是非平凡的——我们能处理的唯一量是对象与它们聚类中心的总距离，而且这个数值是随着组分数目的增长而减小的。相同的问题在混合模型中通过使用对数似然函数 L （和 B 的下界）来消除。图 6-11a 表明混合模型中对数似然函数 L 随着组分的增长而增长。为了理解为什么会出现这种情况，考虑图 6-11b 中使用 $K=10$ 。3 个原始组分（图 6-9e）中的每一个现在被很多更小的组分代替。想象 3 维空间中这些高斯分布的点（就像我们在图 2-8 中做的一样）。因为它们的体积一定等于 1（因为它们是密度函数），所以如果它们占据的区域越小（图 6-11b）中的椭圆越小），那么它们肯定越大。数据集的似然函数，也就是每个数据点高度的一个乘积（或者说对数似然函数是高度对数的和）就会更大。如果我们增加更

多的组分，我们就需要考虑的区域就会更小，因此似然函数就会进一步地变大。

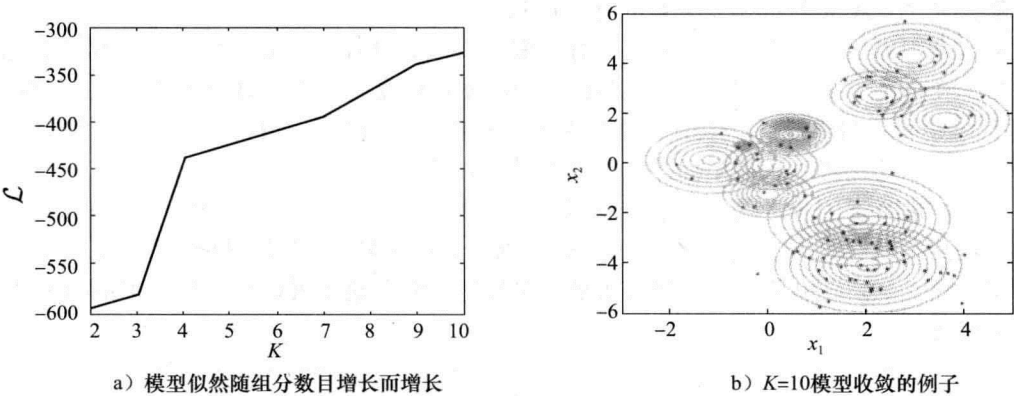


图 6-11 对数似然函数 L 随着组分 K 的增大而增大

幸运的是，我们可以通过一个验证集计算似然函数（例如，交叉验证）来克服这个问题。图 6-12 展示了一个 10 折交叉验证的结果（MAT LAB 脚本：gmixcv.m）。线和条表示验证数据的似然函数的均值和标准差。结果并不是结论性的，比如强烈建议一个特定的组分数目，而是给我们一个可能的数目区域，比如 3~8。在我们的实验中，我们考虑得到尽可能高的精度。得到这样一个类数目的量是它优于 K 均值方法的重要优点，因为目前我们很难在 K 均值中找到任何表明分类数目的指标。

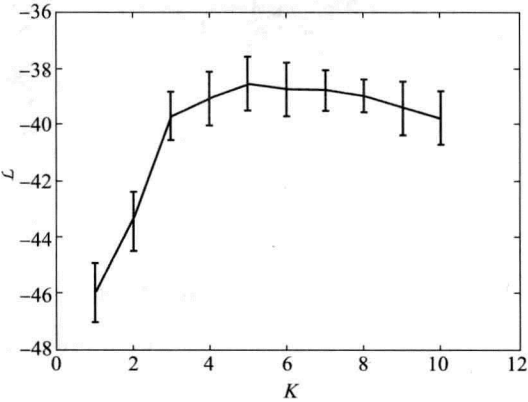


图 6-12 图 6-8 数据中的高斯混合模型的 10 折交叉验证的结果

当然，如果聚类只是我们分析中一步，我们就可以使用其他的度量指标（比如，分类准确率）来选择类的个数。而且，目前迅速发展的非参数方法可以使我们通过马尔科夫链蒙特卡洛方法（类似于第 4 章介绍的 Metropolis-Hastings 方法）来对组分的数目取样。这些方法超出了本书的范围，但我们建议有兴趣的读者阅读本章后面给出的其他阅读材料。

6.3.7 混合组分的其他形式

混合模型超越 K 均值的第二个优势在于它们具有很大的灵活性。具体来讲， $p(x_n|\cdots)$ 可以取任意概率密度形式。在前面的例子中，我们采用高斯（及变换形式）。本节，我们将用一个二值数据集说明经常使用到一些其他的组分形式。但在这之前，有必要花一点时间在高斯上，因为它经常以一些稍加改进的形式出现。

尤其是，由于没有足够的数据用以可靠地估计全部协方差矩阵，通常有必要为混合组分协方差矩阵增加一些限制条件。例如，如果我们得到的是 10 维数据而不是 2 维数据，则将需要更多的数据以能够估计每个协方差矩阵所需要的 55 个参数。为了解决这个问题，通常假设协方差矩阵只有对角元素。这与第 2 章中维度独立的假设是等价的。与 EM 算法唯一不同的是 Σ_k 的更新，它被拆分为对每个 d 维方差 σ_{kd}^2 的更新（见练习 EX 6-1）。一个更加极端的情况是，当协方差假定为各向同性（isotropic）（每一个对角线元素值相同的对角线）时， $\Sigma_k = \sigma_k^2 I$ 。同样，算法唯一的不同在于 Σ_k 的更新（见练习 EX 6-2）。

226
229

现在简要描述对于二值数据的混合模型。每个数据对象 \mathbf{x}_n 为一个二值集合 D 。例如， $D=10$ 维的数据对象可能是：

$$\mathbf{x}_n = [0, 1, 0, 1, 1, 1, 0, 0, 0, 1]$$

图 6-13 显示了一组 10 维数据集的例子，每行表示一个数据对象。假设在特定的组分中维度之间是独立的， $p(\mathbf{x}_n | \dots)$ 可以表示为伯努力分布的乘积（见 2.3.1 节）：

$$p(\mathbf{x}_n | \mathbf{p}_k) = \prod_{d=1}^D p_{kd}^{x_{nd}} (1 - p_{kd})^{1-x_{nd}} \quad (6-20)$$

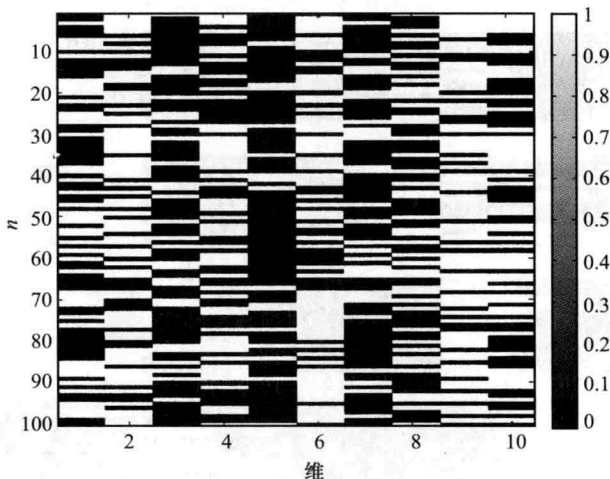


图 6-13 $N=100$ 个对象 $D=10$ 维的二值数据集实例。每行表示一个对象

其中， $\mathbf{p}_k = [p_{k1}, \dots, p_{kD}]^T$ 为第 k 组分指定维的概率向量 ($0 \leq p_{kd} \leq 1$)。与 EM 算法有两点不同。第一，当更新 q_{nk} 时，式 (6-18) 变为：

$$q_{nk} = \frac{\pi_k p(\mathbf{x}_n | \mathbf{p}_k)}{\sum_{j=1}^K \pi_j p(\mathbf{x}_n | \mathbf{p}_j)} \quad (6-21)$$

其中， $p(\mathbf{x}_n | \mathbf{p}_k)$ 由式 (6-20) 给出。第二， \mathbf{p}_k 的更新由 \mathbf{u}_k 和 Σ_k 所代替（式 (6-16) 和式 (6-17)）。

为了得到这个更新，可以从边界 \mathcal{B} （式 (6-8)）中提取数据依赖项。该项则为：

$$\begin{aligned} \mathcal{B} &= \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log p(\mathbf{x}_n | \mathbf{p}_k) + \dots \\ &= \sum_{n=1}^N \sum_{k=1}^K q_{nk} \log \prod_{d=1}^D p_{kd}^{x_{nd}} (1 - p_{kd})^{1-x_{nd}} + \dots \\ &= \sum_{n=1}^N \sum_{k=1}^K q_{nk} \sum_{d=1}^D (x_{nd} \log p_{kd} + (1 - x_{nd}) \log(1 - p_{kd})) + \dots \end{aligned}$$

只保留 p_{kd} 项，得出：

$$\mathcal{B} = \sum_{n=1}^N q_{nk} (x_{nd} \log p_{kd} + (1 - x_{nd}) \log(1 - p_{kd})) + \dots$$

对 p_{kd} 求偏导数，得到：

$$\frac{\partial \mathcal{B}}{\partial p_{kd}} = \sum_{n=1}^N q_{nk} \left(\frac{x_{nd}}{p_{kd}} - \frac{1 - x_{nd}}{1 - p_{kd}} \right)$$

令该式为 0，并整理得到 p_{kd} 的更新。令其为 0 并求解（见练习 EX 6-3），得到：

$$p_{kd} = \frac{\sum_{n=1}^N q_{nk} x_{nd}}{\sum_{n=1}^N q_{nk}} \quad (6-22)$$

230
231

为第 d 维的加权平均, 很类似于高斯混合中 μ_k 的更新。新的 EM 算法包括根据式 (6-21) 更新 q_{nk} (‘E’ 步) 和分别利用式 (6-22)、式 (6-15) 更新 p_k 、 π_k 之间的迭代。与高斯例子一样, 我们需要初始化 π_k 和组分参数, 我们通过设置 $\pi_k = 1/K$ 和为每个 p_{kd} 随机设置 0~1 之间的值。使用 $K=5$ 并运行算法直至收敛, 得到图 6-14 所示的聚类, 其中每个块为一个聚类 (MATLAB 脚本: binmix.m) 我们可以清楚地看到聚类结构——如在聚类 1 (上方) 中, 所有对象在第 9 维中为 1, 第 10、第 7 和第 2 维中为 0。

与这种方法大致相同, 我们可以得出很多其他组分密度的 EM 算法 (见练习 EX 6.6)。

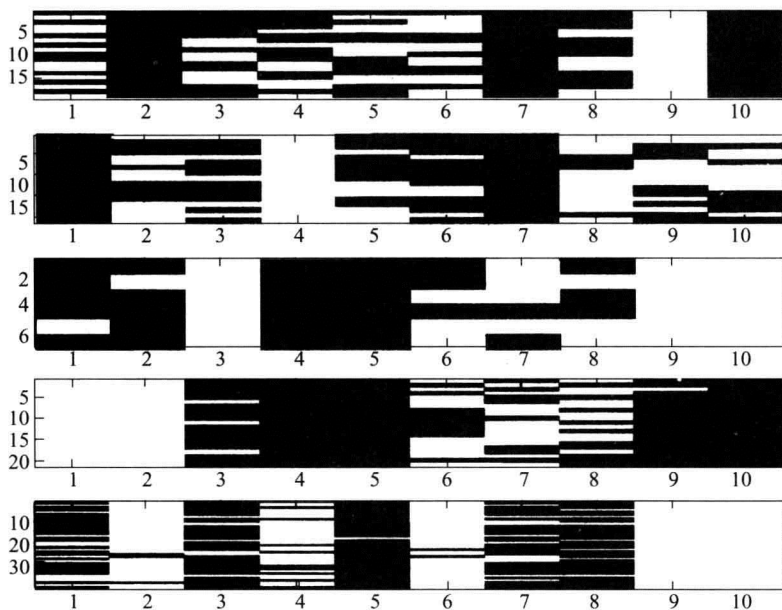


图 6-14 使用二值组分的混合模型从图 6-13 所示数据中抽取的 $K=5$ 个聚类

6.3.8 用 EM 估计 MAP

如果只有有限数量的数据, 那么正规化 EM 所得到的参数估计也许是可行的。直接的方法是将似然与适合的参数先验密度相乘, 得到 MAP 估计 (使后验最大的参数值, 见 4.3 节)。例如, 在上面描述的二值例子中, 我们可能会采用独立 β 先验 (见 2.5.2 节) 用于每个参数 p_{kd} :

232

$$p(p_1, \dots, p_K | \alpha, \beta) = \prod_{k=1}^K \prod_{d=1}^D \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p_{kd}^{\alpha-1} (1 - p_{kd})^{\beta-1}$$

这在边界 \mathcal{B} 中增加了附加 p_{kd} 项。相关项是:

$$\begin{aligned} \mathcal{B} = & (\alpha - 1) \log p_{kd} + (\beta - 1) \log(1 - p_{kd}) \\ & + \sum_{n=1}^N q_{nk} (x_{nd} \log p_{kd} + (1 - x_{nd}) \log(1 - p_{kd})) + \dots \end{aligned}$$

求偏导数, 令其为 0 并用正常方式求解, 得到 (见练习 EX 6.4):

$$p_{kd} = \frac{\alpha - 1 + \sum_{n=1}^N q_{nk} x_{nd}}{\alpha + \beta - 2 + \sum_{n=1}^N q_{nk}} \quad (6-23)$$

注意, $\alpha = \beta = 1$ 时等价于式 (6-22)。正规化效应明显。如果对所有 n 有 $x_{nd} = 1$ 或 $x_{nd} = 0$,

则式 (6-22) 将得到 $p_{kd}=1$ 和 $p_{kd}=0$ 。如果一个新的数据对象没有 $x_{kd}=1$ (或 0), 则属于这个聚类的似然为 0, 而不考虑在其他 $D-1$ 维的值。式 (6-23) 通过对 p_{kd} 进行有效的限制解决该问题, 其最小值和最大值分别为:

$$\frac{\alpha-1}{\alpha+\beta-2}$$

和

$$\frac{\alpha-1+N}{\alpha+\beta-2+N}$$

利用 EM 可以得到很多先验与似然组合的最大后验解。另外的例子见练习 EX 6.5。

6.3.9 贝叶斯混合模型

利用 EM 获得与最大似然或最大后验解对应的点估计, 不是用混合模型进行聚类的唯一方法。具体来讲, 可以使用马尔科夫链蒙特卡罗 (Markov Chain Monte Carlo, MCMC) 方法对聚类分配和相关组分参数进行采样。这有很多优点, 至少事实上它可以解决组分数量确定的问题 (在 6.3.6 节结尾处提到的)。其结果不是一个单一的聚类结果, 而是覆盖聚类结果分布上的一些采样。从纯粹模型角度来讲, 这是一件好事——我们明确地承认这样一个事实, 即聚类 (组分) 数量及其相关分配存在不确定性。在这些不确定性存在的情况下, 坚持一个单一的聚类将伴随其他点估计的所有缺陷。然而, 它带有解释性的问题。对于很多应用, 很难想象如何使用聚类划分的分布, 并且通常人们选择具有最大似然采样的聚类结果。当所需要的最终结果可以表示为与聚类分布相关的期望时, 用 MCMC 方法求混合模型是有帮助的。例如, 如果想要计算两个对象 x_n 和 x_m 处于同一聚类的概率时, 我们可以简单地计数这样的 (属于同一类的) 样本数目除以样本总数。我们无法通过 EM 使用最大似然或 MAP 计算得到这种概率。

233

6.4 小结

本章介绍了聚类的两种算法: K 均值 (包括核 K 均值) 和混合模型。 K 均值的简单性 (和核 K 均值的灵活性) 使其成为一种流行的方法。各种不同的组分模型意味着混合模型 (以及一些变型) 正出现在越来越多的应用中。这些方法也存在一些不足——具体来讲, K 均值算法和混合模型的 EM 算法都只能得到局部最优。换句话讲, 它们得到对应目标函数的极值, 而不能保证是全局最优 (全局最优解)。在这两种方法中, 所得到的解都依赖于初始值—— μ_k 和 Σ_k 的不同随机值会得到不同的聚类结果。

同样需要记住的是, 我们不能在仅仅一章中讲到很多其他的方法。推荐读者进一步研究其他的方法, 例如, 层级聚类 (广泛用于计算生物学)、谱聚类和功能聚类。

6.5 练习

EX 6.1 修改 EM 以更新第 d 维第 k 组分的方差 σ_{kd}^2 , 当聚类组分有对角高斯似然:

$$p(x_n | z_{nk} = 1, \mu_{k1}, \dots, \mu_{kD}, \sigma_{k1}^2, \dots, \sigma_{kD}^2) = \prod_{d=1}^D \mathcal{N}(\mu_{kd}, \sigma_{kd}^2)$$

EX 6.2 使用各项同性的高斯组分重复练习 EX 6.1:

$$p(x_n | z_{nk} = 1, \mu_k, \sigma_k^2) = \prod_{d=1}^D \mathcal{N}(\mu_{kd}, \sigma_k^2)$$

EX 6.3 修改 EM 以更新式 (6-22) 中给出的参数 p_{kd} 表示。

EX 6.4 修改 MAP EM 以更新式 (6-22) 中给出的参数 p_{kd} 表示。假定参数 α 和 β 为 β 先验。

234

EX 6.5 修改 MAP 以更新采用高斯组分满足 D 维独立分布的混合模型:

$$p(\mathbf{x}_n | z_{nk} = 1, \mu_{k1}, \dots, \mu_{kD}, \sigma_{k1}^2, \dots, \sigma_{kD}^2) = \prod_{d=1}^D \mathcal{N}(\mu_{kd}, \sigma_{kd}^2)$$

假设每个 u_{kd} 具有均值 m 和方差 s^2 的独立高斯先验。

EX 6.6 推导适用于混合泊松分布的 EM 算法。假设观测 N 个整数计数, x_1, \dots, x_n 。似然为:

$$p(\mathbf{x} | \Delta) = \prod_{n=1}^N \sum_{k=1}^K \pi_k \frac{\lambda_k^{x_n} \exp\{-\lambda_k\}}{x_n!}$$

235

其他阅读材料

- [1] David Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

这篇文章描述了一种用于文本数据的复杂混合类型的模型。该模型基于一个比本书中描述的更加复杂的生成过程。在机器学习和信息检索文献中表明该特殊模型非常受欢迎。

- [2] Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery*, pages 399–424, 2003.

混合模型可以通过组分密度的任意形式定义。该文中的模型使用了马尔科夫链作为组分密度, 并以一组转移概率作为参数。该模型用来分析互联网浏览行为。

- [3] Guojun Gan, Chaogun Ma, and Jianhong Wu. *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial Mathematics, 2007.

- [4] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31:651–666, 2010.

一篇最近的研究性文献, 给出了关于聚类问题和不同聚类算法的综述。

- [5] Anil K. Jain and R.C. Dubes. *Algorithms For Clustering Data*. Prentice Hall, 1988.

一本关于聚类的教材, 现已绝版, 但可以在作者网站上免费获取: http://www.cse.msu.edu/~jain/Clustering_Jain_Dubes.pdf。

- [6] Anil K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Reviews*, pages 264–323, 1999.

关于聚类技术及在信息检索、图像分割和目标识别等领域中应用讨论的综述。

- [7] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, 2000.

一篇关于统计混合模型的全面描述。

- [8] Carl Rasmussen. The infinite Gaussian mixture model. In *In Advances in Neural Information Processing Systems 12*, pages 554–560, 2000.

最早描述使用狄利克雷过程解决混合模型组分数目的确定问题的文献之一。

237

}

238

主成分分析与隐变量模型

在前面的章节中，我们介绍了两种用于聚类的无监督方法——将数据对象分割成有限数目的不相关组，使得同组中的数据对象具有某些相似性。现在，我们将注意力转移到第二类无监督方法上，该方法通常被归类于投影技术。

本章介绍如何将这些方法用于处理高维的数据集，以及如何通过将数据集投影到低维空间对数据进行可视化或者特征选择。这些技术用于处理更大规模的隐变量模型，我们使用可视化的例子对这个技术领域进行介绍。

7.1 一般问题

本章的出发点是一个包含 N 个对象 \mathbf{y}_n 的数据集。每一个对象是一个 M 维向量。在大多数模型中，参数的数量随着维度 M 的增加而增加。因此，如果 M 很大，那么参数估计是一个挑战性的问题。同样，高维数据也是很难进行可视化的。基于这些原因，将 M 维数据 \mathbf{y}_n 转化为一个 D 维数据 \mathbf{x}_n 通常是很有用的。该过程称为投影。我们将 M 维数据投影到 D 维的同时，希望在某种程度上保留感兴趣的属性。

图 7-1 通过一个更为熟悉的形式对该问题进行说明。图 7-1a、b 给出了一个 3 维对象（手）在 2 维平面上的投影（阴影区域）。图 7-1c 以一个更加数学化的形式展现了投影思想，即将某些 2 维数据（ \mathbf{y}_n ）投影到 1 维空间。该 1 维投影是原有两个维度之一，但是这不是投影的必要条件。为了将图 7-1a 与图 7-1b 相比较，这个原始对象 \mathbf{y}_n 与手相对应， \mathbf{x}_n 与阴影区域相对应。

7.1.1 方差——感兴趣结构的代表

对数据进行投影操作时，我们希望尽可能保留数据中感兴趣的结构。什么是感兴趣的结构呢？图 7-1a、b 都是相同数据的投影，但是，图 7-1a 比图 7-1b 保留了更多原始对象（手）的特点。然而，一般情况下，在投影操作前，我们并不知道原始数据的结构特点，因此也就不能使用该特点优化投影。

在图 7-2a 中，我们使用一个高斯分布生成数据点云团。数据被投影到 A 和 B 两条线上。每条线都是对 2 维数据不同的 1 维表达形式。注意，与图 7-1c 不同，这两条线不对应于任何一个原始维度。这两个 1 维表达形式中的每一个点（任何一条线上的位置）都是一个关于两个原始维度的线性组合。特别地， $x_n = w_1 y_{n1} + w_2 y_{n2}$ （其中， $\mathbf{y}_n = [y_{n1}, y_{n2}]^T$ ），或者，使用向量表达式表示为 $x_n = \mathbf{w}^T \mathbf{y}_n$ ，其中 $\mathbf{w} = [w_1, w_2]^T$ 。

数据在每个 1 维空间上的方差可以通过下式计算：

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N (x_n - \mu_x)^2$$

显然，A 上投影的方差高于 B 上投影的方差。不考虑其他信息，如果数据必须投影到 1 维空间，我们就选择 A。换言之，如果被迫舍弃 A 投影和 B 投影其中的一个，抛弃 B 投影中包含的信息则更加安全可靠。

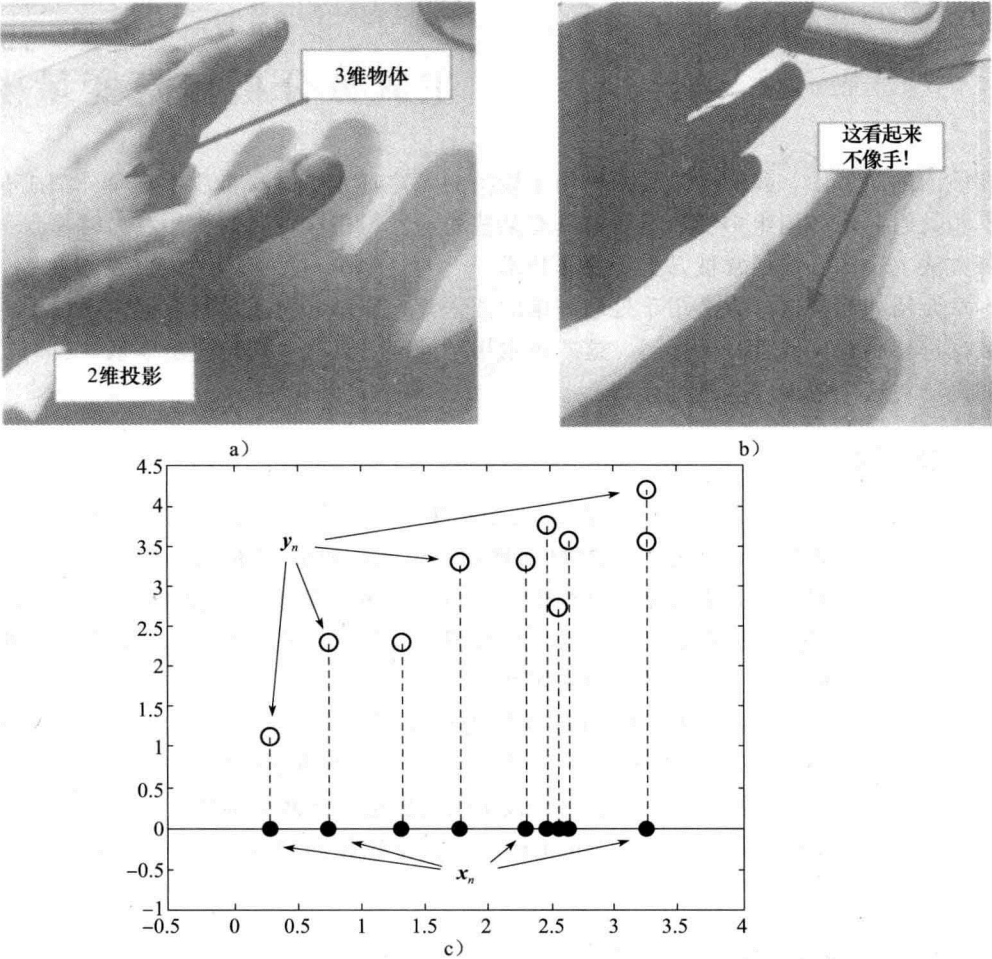


图 7-1 投影的思想。a) 和 b)：手（3 维物体）在灯光下投影到一个 2 维表中（2 维空间）。c) 2 维数据 y_n 投影到 1 维数据空间 x_n 。这里，使用原始坐标轴之一对齐该投影，但这不是必要操作

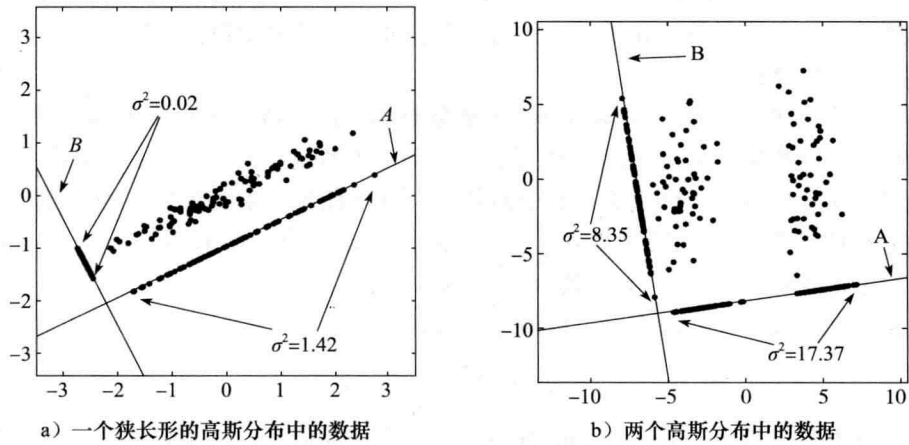


图 7-2 一个例子：两个人造的二维数据集在不同投影方向上的方差。图中给出了这两个方向上的 1 维空间投影（标记为 A 和 B）以及数据在每一个投影上的方差（ σ^2 ）

图 7-2b 中给出一个更加有趣的例子——数据聚类结构的比较。A 投影保持了数据聚类

结构特征,然而B投影没有保持该特征。聚类结构是一个感兴趣的属性,因此A投影比B投影更加让人感兴趣。A投影上数据的方差比B投影上数据的方差大两倍多,这是聚类结构的原因——类内数据点与聚类中心之间距离较大。如果想要表达数据的聚类结构,在使方差最大的方向上对数据进行投影更能保留这一结构特征。

基于这个原因,当考虑投影方向时,方差是一个很好的最大化度量标准。因此,在大多数流行的投影技术中,如主成分分析,使用方差作为最大化度量的标准。

241

7.2 主成分分析

主成分分析(Principal Components Analysis, PCA)可能是目前应用最为广泛的一种统计技术,用于将高维数据投影到低维空间。在机器学习中,该技术大多应用于数据可视化和特征选择。PCA定义了一组线性投影:每个投影维度都是原始数据维度的一个线性组合。即,如果从 M 维投影到 D 维,PCA定义 D 个向量 \mathbf{w}_d (每个都是 N 维的),其中,投影空间中第 d 个元素 x_{nd} (其中 $\mathbf{x}_n = [x_{n1}, \dots, x_{nD}]^T$)使用下式计算:

$$x_{nd} = \mathbf{w}_d^T \mathbf{y}_n$$

因此,学习任务是选择将多少原始数据维度投影到 D ,然后为每一个原始数据维度选择一个投影向量 \mathbf{w}_d 。

PCA使用投影空间上的方差作为选择 \mathbf{w}_d 的准则。特别地, \mathbf{w}_1 是使 x_{n1} 维度上的方差最大的投影向量。同样,第二个投影维度也是根据方差最大化选择 \mathbf{w}_2 ,但是 \mathbf{w}_2 必须垂直于 \mathbf{w}_1 ($\mathbf{w}_1^T \mathbf{w}_2 = 0$)。第三个元素 \mathbf{w}_3 必须满足方差最大化,并且与 \mathbf{w}_1 和 \mathbf{w}_2 同时垂直等。通常:

$$\mathbf{w}_i^T \mathbf{w}_j = 0, \forall j \neq i$$

这组限制条件说明,如果令 $D=M$,那么PCA操作等价于在不损失任何信息的条件下对原始数据进行旋转操作。

另外,PCA强加了一组限制条件,每一个 \mathbf{w}_i 必须定长为1, $\mathbf{w}_i^T \mathbf{w}_i = 1$ 。但是,该条件并没有限制PCA技术本身,更重要的是,仅仅限制了每一个 \mathbf{w} 的方向。

PCA技术的目的是找到投影 $\mathbf{w}_1, \dots, \mathbf{w}_D$,有大量的方法可以应用。本书是通过找到一个表达式为 x_{n1} 的方差来找到这组投影,这也是直接的方法。读者也可以使用统计和机器学习的其他方法获得。

在开始之前,有必要先给出假设条件,即每一个原始数据维度上均值为零:

$$\bar{\mathbf{y}} = \frac{1}{N} \sum_{n=1}^N \mathbf{y}_n = 0$$

该条件可以通过让每一个数据元素 \mathbf{y}_n 减去均值 $\bar{\mathbf{y}}$ 得到满足。

从投影 $D=1$ 维开始。换言之,只需要找到一个 \mathbf{w} 向量。在这种情况下,投影结果是一个标量 x_n 。对于每一个观测值,该值可以通过下式得到:

$$x_n = \mathbf{w}^T \mathbf{y}_n$$

方差 σ_x^2 通过下式得到:

$$\sigma_x^2 = \frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})^2 \quad (7-1)$$

242

基于假设 $\bar{\mathbf{y}}=0$,对表达式进行简化,

$$\begin{aligned} \bar{x} &= \frac{1}{N} \sum_{n=1}^N \mathbf{w}^T \mathbf{y}_n \\ &= \mathbf{w}^T \left(\frac{1}{N} \sum_{n=1}^N \mathbf{y}_n \right) \end{aligned}$$

$$= \mathbf{w}^T \bar{\mathbf{y}} = 0.$$

式 (7-1) 变换为:

$$\sigma_x^2 = \frac{1}{N} \sum_{n=1}^N x_n^2$$

代入 x_n 的定义, 得:

$$\begin{aligned} \sigma_x^2 &= \frac{1}{N} \sum_{n=1}^N (\mathbf{w}^T \mathbf{y}_n)^2 \\ &= \frac{1}{N} \sum_{n=1}^N \mathbf{w}^T \mathbf{y}_n \mathbf{y}_n^T \mathbf{w} \\ &= \mathbf{w}^T \left(\frac{1}{N} \sum_{n=1}^N \mathbf{y}_n \mathbf{y}_n^T \right) \mathbf{w} \\ \sigma_x^2 &= \mathbf{w}^T \mathbf{C} \mathbf{w} \end{aligned} \quad (7-2)$$

其中, \mathbf{C} 是样本的协方差矩阵, 定义为:

$$\mathbf{C} = \frac{1}{N} \sum_{n=1}^N (\mathbf{y}_n - \bar{\mathbf{y}})(\mathbf{y}_n - \bar{\mathbf{y}})^T$$

其中, 假定 $\bar{\mathbf{y}}=0$ 。注意, 该式意味着通过强制 $\bar{\mathbf{y}}$ 为零, 可以对数据进行变换而不损失任何信息。即无论是否强制 $\bar{\mathbf{y}}=0$, \mathbf{C} 都相同。

学习的目标是找到使 \mathbf{w} 最大的 σ^2 值, 所以还要最大化 $\mathbf{w}^T \mathbf{C} \mathbf{w}$ 。这可以通过增加 \mathbf{w} 中元素的值, 进而增加 $\mathbf{w}^T \mathbf{C} \mathbf{w}$ 来实现, 这也是强制 \mathbf{w} 的长度为 1 的原因, $\mathbf{w}^T \mathbf{w} = 1$ 。与第 5 章 SVM 优化和第 6 章 EM 推导一样, 本章使用拉格朗日项 (参见注解 5.1) 将该限制条件与优化策略相结合。特别地, 找到 \mathbf{w} , 使下式最大化:

$$L = \mathbf{w}^T \mathbf{C} \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{w} - 1)$$

对 \mathbf{w} 求偏导, 并使偏导数为零, 重写为

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}} &= 2\mathbf{C}\mathbf{w} - \lambda\mathbf{w} = \mathbf{0} \\ \mathbf{C}\mathbf{w} &= \lambda\mathbf{w} \end{aligned} \quad (7-3)$$

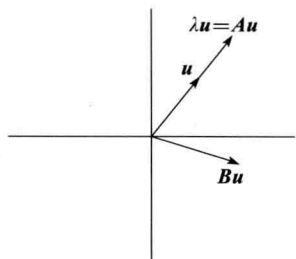
243 (其中我们将系数 2 结合到 λ 常数中)。

注解 7.1 (特征向量与特征值): 方阵 \mathbf{A} 的特征向量/特征值方程为:

$$\lambda_i \mathbf{u}_i = \mathbf{A} \mathbf{u}_i \quad (7-4)$$

该方程的解是一组特征向量 \mathbf{u}_i /特征值 λ_i 对。

右图给出了该方程解的直观解释。一个 M 维向量 \mathbf{u} 乘以一个 $M \times M$ 矩阵 \mathbf{B} 得到另一个 M 维向量。因此, 矩阵 \mathbf{B} 可以看做是对向量 \mathbf{u} 的旋转操作。不同矩阵 \mathbf{B} 对应的旋转操作也不同。对于一个给定的矩阵 \mathbf{A} , 式 (7-3) 中的解是对向量 \mathbf{u} 进行 \mathbf{A} 中定义的旋转操作, 并得到 \mathbf{u} 长度的变化, 该变化的大小用标量 λ 表示。



一般情况下, 如果矩阵 \mathbf{A} 有 M 行 M 列, 则有 M 个特征向量/特征值对成为式 (7-4) 的解。 M 个特征向量相互垂直。本章无法对特征向量/特征值方程的求解过程做更细致的讨论, 但是该解是简单易得的, 例如, 使用 MATLAB 中的 `eigs` 函数。

式 (7-3) 的形式极为常见, 即特征向量/特征值方程 (参见注解 7.1)。比较式 (7-3) 和式 (7-4), 可以看出方差最大化方向上的投影 \mathbf{w} 是协方差矩阵 \mathbf{C} 的一组特征向量。然而, 现在有 M 个, 怎样选择使方差最大的特征向量呢? σ_x^2 的表达式是:

$$\sigma_x^2 = \mathbf{w}^T \mathbf{C} \mathbf{w}$$

回顾 $\mathbf{w}^T \mathbf{w} = 1$, 因此等式左边乘以 $\mathbf{w}^T \mathbf{w}$:

$$\sigma^2 \mathbf{w}^T \mathbf{w} = \mathbf{w}^T \mathbf{C} \mathbf{w}$$

两边同时除以 \mathbf{w}^T , 剩余项类似于式 (7-3):

$$\sigma^2 \mathbf{w} = \mathbf{C} \mathbf{w}$$

该式给出了特征向量/特征值对 (λ, \mathbf{w}) , 特征值 λ 对应于 \mathbf{w} 定义的投影空间中数据的方差。如果找到协方差矩阵 \mathbf{C} 中的 M 个特征向量/特征值对, 则其中最大的特征值对应的特征向量/特征值对就是方差最大化方向上的投影 \mathbf{w}_1 。排在第二位的特征值对应 \mathbf{w}_2 , 第三位对应 \mathbf{w}_3 等。

总之, 在数据对象集合 $\mathbf{y}_1, \dots, \mathbf{y}_N$ 上进行投影操作包括以下步骤 (括号中的表达式是对应的矩阵操作, 这里定义 $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]^T$):

244

1) 让每一个元素减去 $\bar{\mathbf{y}}$, 使 M 维数据的均值为 0, 其中 $\bar{\mathbf{y}} = \frac{1}{N} \sum_{n=1}^N \mathbf{y}_n$ 。

2) 计算样本的协方差矩阵 $\mathbf{C} = \frac{1}{N} \sum_{n=1}^N \mathbf{y}_n \mathbf{y}_n^T$ 或者 $\mathbf{C} = \frac{1}{N} \mathbf{Y}^T \mathbf{Y}$ 。

3) 求解得到 M 个特征向量/特征值对。该求解操作可以通过 MATLAB 中的 `eigs` 函数完成。

4) 找到 D 个最大特征值对应的特征向量 $\mathbf{w}_1, \dots, \mathbf{w}_D$ 。

5) 在投影空间中, 为对象 n 建立第 d 维 $x_{nd} = \mathbf{w}_d^T \mathbf{y}_n$ (或者 $\mathbf{X} = \mathbf{Y} \mathbf{W}$, 其中 $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_D]$, 即按顺序排列 D 个特征向量建立 $M \times D$ 矩阵, \mathbf{X} 是 $N \times D$ 矩阵, 定义为 $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$)。

为了说明这一点, 回到图 7-2 的简单例子中。在这些数据点中, 选择的投影方向就是主成分。例如, 在 2 维空间中, 存在一个包含两个成分的最大值 (2×2 的协方差矩阵只有两个特征向量。在 2 维空间中, 不可能有更多的相互垂直的方向)。对于这些点, 前文提到该过程只需定义在其上投影的直线的方向。为了看得更加清晰, 这里将直线向下移动一点 (A 投影), 向左移动一点 (B 投影)。

图 7-3 是一个更复杂的例子 (MATLAB 脚本: `pcaexample.m`)。建立一个数据集, 其中每个数据对象属于 3 个聚类之一 (参见图 7-3a)。然后通过添加额外的 5 个维度将数据变得更复杂一些, 每个维度的值从 $\mathcal{N}(0, 1)$ 依据下式采样获得:

$$y_{nd} \sim \mathcal{N}(0, 1), d = 3, \dots, 7, n = 1, \dots, N$$

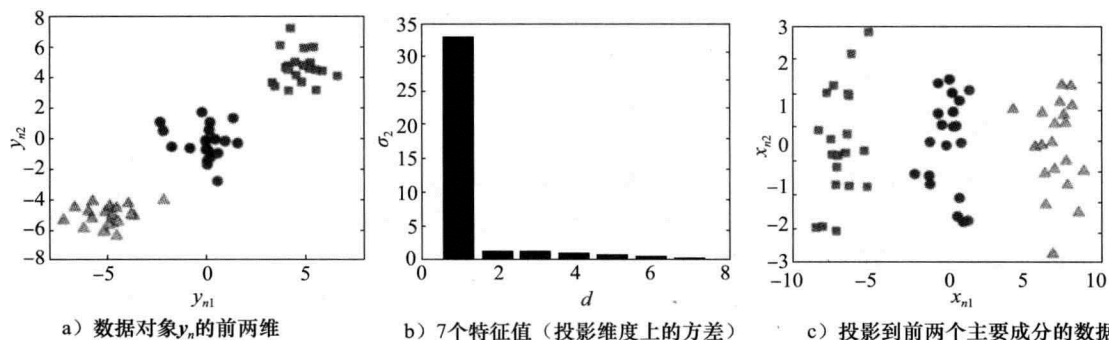


图 7-3 只需一个投影维度的 PCA 例子。数据对象 \mathbf{y}_n 具有 7 个维度。前两维保留了 a 中所示的聚类结构。剩余维度的值从 $\mathcal{N}(0, 1)$ 中随机选择

换言之, 数据的结构化信息只在前两个维度中存在, 而额外的 5 个维度是噪声数据。这种情况在现实中是存在的, 即很多数据对象的不同属性是可以观察记录的, 但是它们的先验知识却很难得到。对数据进行均值中心化后, 图 7-3b 中给出了协方差矩阵 $\mathbf{C} = \mathbf{Y}^T \mathbf{Y}$ 中这 7 个

245

维度的特征值，并根据大小排序。前文提到这些特征值相当于 D 个潜在投影维度上的方差。可以看出，最大的特征值远远大于其他特征值。因此，只需要一个投影维度就可以捕获 7 个维度上的大部分方差信息。这似乎有些不寻常，因为原始聚类结构在前两个维度上。但是，回到图 7-3a 中，可以看出该聚类结构实际上只是 1 维空间上的，因为聚类在直线 $y_{n1} = y_{n2}$ 上。使用一个投影维度对其进行描述就足够了。虽然在前两个投影维度上画出这些点看起来更加清晰（见图 7-3c），但第一个投影维度 x_{n1} 保存了全部的聚类结构信息。

该例表明了 PCA 的一个重要特征。特征谱（特征值的重要性，见图 7-3b）显示了数据中感兴趣特征的数量。例如，图 7-3b 显示了使用两个维度并不会比使用一个维度获得更多的信息。图 7-4 给出第二个例子（MATLAB 脚本：pcaexample2.m）。图 7-4a 给出一个不同的例子，使用前两维保存聚类结构信息（其他 5 维构造与之前例子相同）。在这个例子中，一个方向上有 4 个聚类，只用一个维度无法表示。使用一个单独的线性投影无法分离所有的聚类。因此，投影操作需要多个特征值。图 7-4b 给出特征值的证明——前两个维度的特征值远远大于其余 5 个维度。从图 7-4c 中可以看到数据投影到前两个成分的结果，很明显，缩减后的投影空间中保留了该聚类结构。

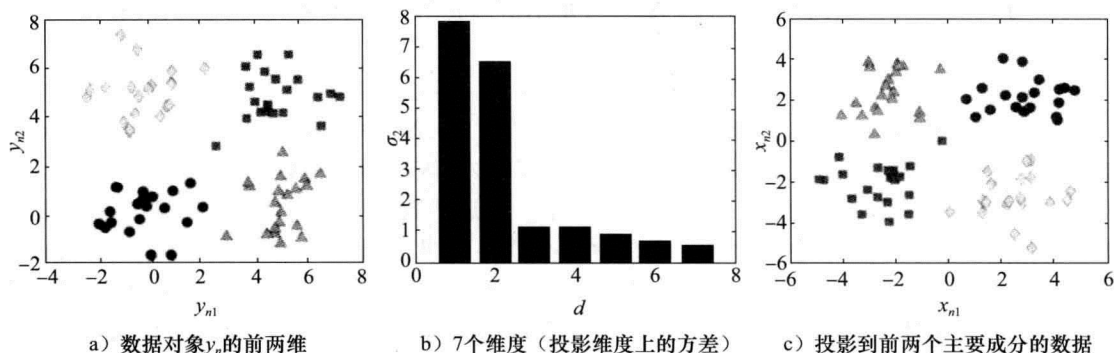


图 7-4 需要投影到两个维度的 PCA 例子

有必要重申这些例子中的要点。首先，前文提到的这些例子中都增加了 5 个“随机”的维度。因此，不同于图 7-3a 和图 7-4a，该问题具有实际意义。其次，虽然使用不同符号（例如，圆圈、方块等）对数据对象进行标记，但该标记信息不会用于 PCA——无监督学习。最后，聚类结构与所在的前两个维度的位置无关。如果对 \mathbf{Y} 的列随意排序（例如，对 \mathbf{Y} 的维度重新排序），结果应当是一样的。

7.2.1 选择 D

前文使用特征谱（和数据知识）来表示投影维度的数量。一般情况下，对 D 的选择根据应用不同而不同。例如，如果在可视化任务中使用 PCA 来对高维数据进行可视化，可视化的维度需要从实践的角度加以限制：最多不超过 3 维。

针对其他应用，特征谱同样提供了有用的信息。但是，其具体解释很大程度上依赖于主观思想（这种信息不能像图 7-3b 和图 7-4b 中一样清晰可见）。如果 PCA 是大型学习系统的一部分，其客观度量是很重要的。例如，PCA 常用于分类任务之前的特征提取。假设图 7-4 是一个真实的四类分类问题，该问题由数据矩阵 \mathbf{Y} 和标记 \mathbf{t} 组成，使用投影后的数据 \mathbf{X} 而不是原始数据 \mathbf{Y} 进行分类是合理的。该问题中， D 值应当使用其他方法更好地进行选择，从而使分类效果更好，例如，交叉验证。

7.2.2 PCA 的局限性

虽然 PCA 已经成功应用于很多领域，但是与所有模型相同，PCA 具有明显的局限性。特别地，PCA 对数据有以下两个假设：

- 1) 数据必须是实值。
- 2) 数据没有缺失值。

在很多问题中，数据满足这两个假设条件，但是很多却不能满足。例如，数据缺失值是一个科学数据中经常出现的问题，因为数据的质量是动态变化的，并且难以通过实验设备加以控制。购买记录的数据集（某人是否购买了某些物品）可能是一个二值数据而非实值或连续值。一个明显不满足这两个条件的例子是电影等级数据集，假定存在一个矩阵，每一行代表观看者，每一列代表电影。第 i 行第 j 列是第 i 个观看者对第 j 个电影的评价等级。一般情况下，该值是一个整数（0~5 星级，不是实值或连续值），而且让一个观看者观看每一部电影并评出等级是很不现实的，所以有很多缺失值。

本章从经典 PCA 的局限性出发，介绍隐变量模型以及如何使用隐变量模型进行学习和推理。需要强调的是，在 PCA 领域之外还有很多种隐变量模型——例如，第 6 章介绍的混合模型。但是，从经典 PCA 的局限性出发是一个很好的路线，沿着这个有利条件，本章介绍变分推理——一种获得难以求出的后验密度函数的近似方法。近年来，该方法在机器学习领域十分流行，因为它的实际性能好并且计算代价低。

247

7.3 隐变量模型

在很多实际应用中，感兴趣对象的特征并未体现在观测数据中。这些隐变量（也称隐藏变量）可以分为以下两类：

- 1) 变量对应于对象的真实特征，但该特征无法被量化（例如，可能是无法量化的技术特征）。
- 2) 抽象的质量，该质量在现实中不存在，但是在模型中假定其存在且起作用。

生物数据分析领域存在很多上述情况。假定一个生物系统，该系统有 3 个分子类型：A、B 和 C。A 和 C 是很容易量化的，但 B 是隐藏的。在这种情况下，B 可以视为模型中的隐变量，其值通过对 A 和 C 中的数据学习得到。

本章侧重于讨论第二类隐变量。PCA 是一个很好的例子——观察一组 M 维的向量 \mathbf{y}_n ，并用其建立一个 D 维向量集合 \mathbf{x}_n 。其中，输入向量可能是现实存在的真实事物的量化结果。然而， \mathbf{x}_n 是隐变量，模型中的隐变量是基于假设建立的——不需要真实存在。建立的目的是隐变量比原始变量更加有用，例如，可以进行可视化。

根据本章主题，下面集中介绍类似于 PCA 的模型。首先，前文讲过这些模型的用处，然后将它放在隐变量模型框架下。

7.3.1 隐变量模型中的混合模型

在 6.3 节中，混合模型作为一种强有力的聚类技术已经介绍过。在一组数据的生成过程中，针对每个数据对象，目的是从 K 个可能的类别中选择一个，并从这个类别中采样这个数据对象。前文引入一组指示变量 z_{nk} ，如果第 n 个对象是由第 k 个类别产生的，则 $z_{nk}=1$ 。这些指示变量就是隐变量（现实中不一定存在），用于建立混合模型。当使用隐变量对混合模型中的参数进行推理时，尽管式 (6-19) 给出的参数 q_{nk} 可以使用后验概率进行解释，但是 z_{nk} 没有被精确使用。其中后验概率是对象 n 经由第 k 个类别产生的概率， $p(z_{nk}=1 | \mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\Delta})$ 。该模型定义暗示可以通过生成一组隐变量，并使用它们的取值进行学习。

248

7.3.2 小结

本章开头, PCA 被当做一种将 M 维数据投影到 D 维空间 (其中 $D < M$) 中的工具加以介绍, 该技术可用于可视化过程 (其中, $D \leq 2$), 或者通用的无监督预处理工具, 该工具用于其他数据分析之前 (例如, 分类、聚类等)。PCA 有一些缺点 (只能处理实值或连续值数据, 不能处理缺失数据), 这些缺点也是在随后章节中需要处理的。在这种情况下, PCA 是隐变量模型家族中的一种, 第 6 章介绍的混合模型也属于这一家族。为了在概率 PCA 模型下进行推理, 需要使用近似技术。第 4 章已经介绍了这类方法, 本章介绍另一种技术, 变分贝叶斯。

7.4 变分贝叶斯

变分贝叶斯 (Variational Bayes, VB) 使用近似推理技术, 该技术在机器学习领域十分流行, 因为它具有很好的实际性能和较低的计算代价。与 4.4 节中介绍的拉普拉斯推理相类似, 通过该技术可以轻易地获得较难的后验概率, 其中近似后验中的参数是被优化的, 目的是使近似值尽可能与真实值一致。

虽然 VB 用于建立近似后验, 这不是它的主要目的。其后验的近似值需要通过最大化对数边缘似然函数获得。

通常情况下, 初始条件是一组数据 \mathbf{Y} 和一个含有参数/隐变量 $\boldsymbol{\theta}$ 的模型。注意, 这里将模型的所有参数和隐变量用相同的符号 ($\boldsymbol{\theta}$) 表示。因为在贝叶斯框架下, 如何处理隐变量和模型参数没有太大区别: 它们都是未知的, 并当作随机变量处理。边缘似然 $p(\mathbf{Y})$ 定义为:

$$p(\mathbf{Y}) = \int p(\mathbf{Y}, \boldsymbol{\theta}) d\boldsymbol{\theta} \quad (7-5)$$

在式 (7-5) 中, 忽略固定的边界条件, 包括模型类型、先验参数、超参数等。这些条件随模型/问题的不同而不同, 因此这里使用一般形式的表达式, 当处理特定问题时再进行特殊化。该式 (7-5) 的一种特殊形式已经在 3.4 节中第一次介绍边缘似然时已经提到过。

注意, 该表达式一般也将联合概率密度 $p(\mathbf{Y}, \boldsymbol{\theta})$ 写成其组成部分表示形式:

$$p(\mathbf{Y}) = \int p(\mathbf{Y}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

因此, 式 (7-5) 中计算的边缘似然是所有参数值 (和隐变量) 的似然 $p(\mathbf{Y}|\boldsymbol{\theta})$ 的平均结果, 结果是使用先验知识进行加权的 $p(\boldsymbol{\theta})$ 。该表达式可以在边界条件 (模型结构、先验参数) 上最大化。不幸的是, 最大化该式是非常困难的, 因为取值范围是一个潜在的高维参数空间。该问题的一种处理方式是降低对数边缘似然的边界, 该处理方式类似于第 6 章的 EM 算法推导中所使用的方法。这里使用詹森不等式 (参见式 (6-7)):

$$\log \mathbf{E}_{p(z)} \{f(z)\} \geq \mathbf{E}_{p(z)} \{\log f(z)\}$$

对数边缘似然为:

$$\log p(\mathbf{Y}) = \log \int p(\mathbf{Y}, \boldsymbol{\theta}) d\boldsymbol{\theta}$$

从等式右侧引入一个关于 $\boldsymbol{\theta}$ 的任意分布 $Q(\boldsymbol{\theta})$ 开始:

$$\log p(\mathbf{Y}) = \log \int Q(\boldsymbol{\theta}) \frac{p(\mathbf{Y}, \boldsymbol{\theta})}{Q(\boldsymbol{\theta})} d\boldsymbol{\theta}$$

从式 (6-7) 中的詹森不等式可知, 一个期望的对数值永远比对数的期望值大。等式右侧可写为 $Q(\boldsymbol{\theta})$ 的一个期望 (其中 $p(\mathbf{Y}, \boldsymbol{\theta})/Q(\boldsymbol{\theta})$), 因此使用詹森不等式建立下界 $\mathcal{L}(Q)$:

$$\begin{aligned} \log p(\mathbf{Y}) &= \log \int Q(\boldsymbol{\theta}) \frac{p(\mathbf{Y}, \boldsymbol{\theta})}{Q(\boldsymbol{\theta})} d\boldsymbol{\theta} \\ &\geq \int Q(\boldsymbol{\theta}) \log \frac{p(\mathbf{Y}, \boldsymbol{\theta})}{Q(\boldsymbol{\theta})} d\boldsymbol{\theta} = \mathcal{L}(Q) \end{aligned} \quad (7-6)$$

计算真实的对数边缘似然与新下界之间的差值, 该差值反映了如何获得一个近似后验:

$$\begin{aligned}
 \log p(\mathbf{Y}) - \mathcal{L}(\mathbf{Q}) &= \log p(\mathbf{Y}) - \int \mathbf{Q}(\boldsymbol{\theta}) \log \frac{p(\mathbf{Y}, \boldsymbol{\theta})}{\mathbf{Q}(\boldsymbol{\theta})} d\boldsymbol{\theta} \\
 &= \log p(\mathbf{Y}) - \int \mathbf{Q}(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta}|\mathbf{Y})p(\mathbf{Y})}{\mathbf{Q}(\boldsymbol{\theta})} d\boldsymbol{\theta} \\
 &= \log p(\mathbf{Y}) - \int \mathbf{Q}(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta}|\mathbf{Y})}{\mathbf{Q}(\boldsymbol{\theta})} d\boldsymbol{\theta} - \int \mathbf{Q}(\boldsymbol{\theta}) \log p(\mathbf{Y}) d\boldsymbol{\theta} \\
 &= \log p(\mathbf{Y}) - \int \mathbf{Q}(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta}|\mathbf{Y})}{\mathbf{Q}(\boldsymbol{\theta})} d\boldsymbol{\theta} - \log p(\mathbf{Y}) \int \mathbf{Q}(\boldsymbol{\theta}) d\boldsymbol{\theta} \\
 &= \log p(\mathbf{Y}) - \int \mathbf{Q}(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta}|\mathbf{Y})}{\mathbf{Q}(\boldsymbol{\theta})} d\boldsymbol{\theta} - \log p(\mathbf{Y}) \\
 \log p(\mathbf{Y}) - \mathcal{L}(\mathbf{Q}) &= - \int \mathbf{Q}(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta}|\mathbf{Y})}{\mathbf{Q}(\boldsymbol{\theta})} d\boldsymbol{\theta} = -\text{KL}[\mathbf{Q}(\boldsymbol{\theta}) \parallel p(\boldsymbol{\theta}|\mathbf{Y})] \quad (7-7)
 \end{aligned}$$

该表达式就是真实后验 $p(\boldsymbol{\theta}|\mathbf{Y})$ 和近似后验 $\mathbf{Q}(\boldsymbol{\theta})$ 之间的 Kullback-Leibler (KL) 散度, 参见注解 7.2。

250

注解 7.2 (Kullback-Leibler 散度): 度量两个概率分布之间的不同是很重要的。例如, 如果想要找到与真实后验相似的近似后验, 首先需要定义什么是相似性! Kullback-Leibler 散度就是这样一个用于衡量相似性的标准, 该标准起源于变分贝叶斯技术。它定义离散与连续分布:

$$\text{KL}[q(x) \parallel p(x)] = \int q(x) \log \frac{p(x)}{q(x)} dx \quad (\text{连续})$$

$$\text{KL}[q(x) \parallel p(x)] = \sum_x q(x) \log \frac{p(x)}{q(x)} \quad (\text{离散})$$

连续分布通常是很难计算的, 因为需要在一个潜在的高维空间中计算积分。

Kullback-Leibler (KL) 散度的一个重要属性是其不对称性—— $\text{KL}[q(x) \parallel p(x)] \neq \text{KL}[p(x) \parallel q(x)]$ 。Kullback-Leibler 散度通常小于等于 0, 当 $p(x)=q(x)$ 时为最大值。

式 (7-7) 左侧必须大于或等于 0 (记得 $\mathcal{L}(\mathbf{Q})$ 是 $\log p(\mathbf{Y})$ 的下界)。Kullback-Leibler 散度是对两个概率分布之间不相似性的一个度量, 只有当这两个分布完全相同时, 取值为 0; 否则, 小于 0。使用不同的 \mathbf{Q} 最大化 $\mathcal{L}(\mathbf{Q})$, 减少了 KL 散度为负的可能性, 因此使得 $\mathbf{Q}(\boldsymbol{\theta})$ 越来越接近于真实后验 $p(\boldsymbol{\theta}|\mathbf{Y})$ 。如果 $\mathbf{Q}(\boldsymbol{\theta})$ 和 $p(\boldsymbol{\theta}|\mathbf{Y})$ 相同, 该下界等于真实对数边缘似然 (参见练习 EX 7.1)。

7.4.1 选择 $\mathbf{Q}(\boldsymbol{\theta})$

前文提到, 如果通过最大化 $\mathbf{Q}(\boldsymbol{\theta})$ 的下界使 $\mathbf{Q}(\boldsymbol{\theta})$ 越来越接近于真实后验, 那么首先要选择 $\mathbf{Q}(\boldsymbol{\theta})$ 的形式, 并且该形式的选择可以简化最大化式 (7-6) 边界的过程。但这是有利有弊的, $\mathbf{Q}(\boldsymbol{\theta})$ 的表达式越复杂, 下界就越难优化, 但可以得到一个较好的近似结果。 $\mathbf{Q}(\boldsymbol{\theta})$ 的形式简单, 优化过程也就简单, 但近似结果可能很差。通常的假设条件是不同参数/隐变量 $\boldsymbol{\theta}$ 之间的独立性假设:

$$\mathbf{Q}(\boldsymbol{\theta}) = \prod_{l=1}^L \mathbf{Q}_l(\boldsymbol{\theta}_l) \quad (7-8)$$

其中, $l=1 \cdots L$ 是相互独立的或者参数集合或者隐变量集合。例如, 模型可能有 M 个参数向量 \mathbf{w}_m 和 N 个隐变量 \mathbf{x}_n , 分别表示为 \mathbf{W} 和 \mathbf{X} 。假设这些参数集合之间是相互独立的:

251

$$Q(\mathbf{W}, \mathbf{X}) = Q_W(\mathbf{W})Q_X(\mathbf{X})$$

进一步, 假设该分布中一个或者多个 M (或者 N) 类别是独立的:

$$Q_W(\mathbf{W}) = \prod_{m=1}^M Q_{w_m}(\mathbf{w}_m), \quad Q_X(\mathbf{X}) = \prod_{n=1}^N Q_{x_n}(\mathbf{x}_n)$$

更进一步, 例如, 我们假设在 \mathbf{x}_n 的 D 维是独立的:

$$Q_X(\mathbf{X}) = \prod_{n=1}^N \prod_{d=1}^D Q_{x_{nd}}(x_{nd})$$

假设在真实后验中这些参数是相互依赖的, 独立性假设越多, 近似结果可能越差。这也是上文的一个例子: 较多的独立性假设使得优化下界变得容易, 但近似结果变差。

7.4.2 优化边界

如果使用式 (7-8) 来建立 $Q(\boldsymbol{\theta})$, 该边界可使用下面形式的分布优化获得:

$$Q_l(\boldsymbol{\theta}_l) = \frac{\exp(\mathbf{E}_{k \neq l} \{\log p(\mathbf{Y}, \boldsymbol{\theta})\})}{\int \exp(\mathbf{E}_{k \neq l} \{\log p(\mathbf{Y}, \boldsymbol{\theta})\}) d\boldsymbol{\theta}_l} \quad (7-9)$$

其中期望是式 (7-8) 中除了第 l 个分布外, 所有独立分布之和。

这个表达式并不像看起来那么复杂。分母是简单的归一化常数, 通常情况下, 其形式由分子中 $\boldsymbol{\theta}_l$ 的形式决定。例如, 线性 ($\boldsymbol{\theta}_l^T \mathbf{b}$) 和多项式 ($\boldsymbol{\theta}_l^T \mathbf{A} \boldsymbol{\theta}_l$) 项决定了 $Q_l(\boldsymbol{\theta}_l)$ 是高斯的, 那么归一化常数就确定了。

计算每一个 $Q_l(\boldsymbol{\theta}_l)$ 需要对所有 $Q_k(\boldsymbol{\theta}_k)$ 取期望。类似于第 6 章的 EM 算法, 近似后验的优化通过迭代过程完成。

7.5 PCA 的概率模型

为了说明变分贝叶斯, 这里从一个类似于 PCA 的概率模型开始。假定观察到 $n=1 \cdots N$ 个 M 维输入向量 \mathbf{y}_n , 任务是找到一个 D 维表示 \mathbf{x}_n (其中 $D < M$)。首先, 使用下面模型对 \mathbf{y}_n 和 \mathbf{x}_n 建立联系:

$$\mathbf{y}_n = \mathbf{W} \mathbf{x}_n + \mathbf{v}$$

252

其中, \mathbf{W} 是一个 $M \times D$ 的矩阵且 \mathbf{v} 是一个 $M \times 1$ 的噪声向量。这个模型的图形表示 (参见 3.6 节) 如图 7-5 所示。对先验进行以下假设:

$$p(\mathbf{x}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$$

$$p(\mathbf{W}) = \prod_{m=1}^M p(\mathbf{w}_m)$$

$$p(\mathbf{w}_m) = \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$$

$$p(\mathbf{y}_{nm}) = \mathcal{N}(\mathbf{w}_m^T \mathbf{x}_n, \tau^{-1})$$

$$p(\tau | a, b) = \Gamma(a, b) = \frac{b^a \tau^{a-1} e^{-b\tau}}{\Gamma(a)}$$

其中 $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_M]^T$, 为了方便起见, 这里对噪声 $\tau(\tau^{-1} = \sigma^2)$ 的参数定义使用精度而不是方差。

任务是使用变分贝叶斯推理 $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$ 、 \mathbf{W} 和 τ 的近似后验。第一步是对 $Q(\mathbf{W}, \mathbf{X}, \tau)$ 进行分解:

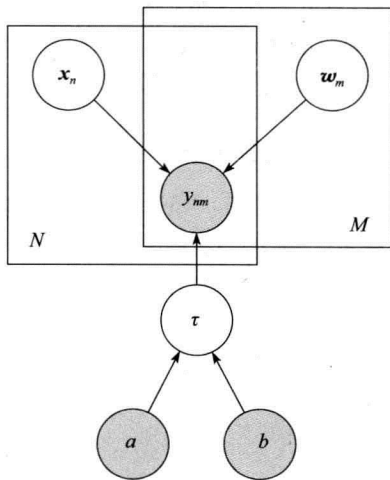


图 7-5 概率 PCA 模型的图形表示

$$Q(W, X, \tau) = Q_\tau(\tau) \left[\prod_{n=1}^N Q_{x_n}(x_n) \right] \left[\prod_{m=1}^M Q_{w_m}(w_m) \right]$$

即, 假设这3个参数集合是相互独立的 (为了简单起见, 从现在开始, 将隐变量 x_n 作为参数), 并额外添加 X 和 W 的不同向量组分之间的独立性。

为了获得每一个 $Q_i(\theta_i)$ 的表达式, 需要从式 (7-9) 中求取 $\log p(Y, \theta)$ 的期望值。在这个例子中, 该期望值是 (在标准 IID 假设条件下):

$$\begin{aligned} p(Y, X, W, \tau) &= p(\tau|a, b) \left[\prod_{m=1}^M p(w_m) \right] \left[\prod_{n=1}^N p(x_n) p(y_n | W, x_n, \tau) \right] \\ \log p(Y, X, W, \tau) &= \log p(\tau|a, b) + \sum_{m=1}^M \log p(w_m) + \sum_{n=1}^N \log p(x_n) \\ &\quad + \sum_{n=1}^N \log p(y_n | W, x_n, \tau) \end{aligned}$$

因为噪声向量 v 的协方差矩阵是对角协方差矩阵, 所以等式右侧的最终形式可以展开为每一个中独立元素 y_n 的相加:

$$\begin{aligned} \log p(Y, X, W, \tau) &= \log p(\tau|a, b) \\ &\quad + \sum_{m=1}^M \log p(w_m) \\ &\quad + \sum_{n=1}^N \log p(x_n) \\ &\quad + \sum_{n=1}^N \sum_{m=1}^M \log p(y_{nm} | w_m, x_n, \tau) \end{aligned} \quad (7-10)$$

其中

$$p(y_{nm} | w_m, x_n, \tau) = \mathcal{N}(w_m^T x_n, \tau^{-1})$$

将以此定义这个后验概率中的每一项。

7.5.1 $Q_\tau(\tau)$

从式 (7-9) 可得:

$$Q_\tau(\tau) \propto \exp(E_{Q_X(X)Q_W(W)} \{ \log p(Y, X, W, \tau) \})$$

忽略表达式中任何不包含 τ 的项, 因为这些项包含在归一化常数中。式 (7-10) 中依赖于 τ 的项只有第一项和最后一项。重写为:

$$\begin{aligned} \log p(Y, X, W, \tau) &\propto a \log b + (a-1) \log \tau - b\tau - \log \Gamma(a) \\ &\quad - \frac{NM}{2} \log 2\pi + \frac{NM}{2} \log \tau - \frac{\tau}{2} \sum_n \sum_m (y_{nm} - w_m^T x_n)^2 \end{aligned}$$

去掉新的不包含 τ 的项 (这里有 $\exp(A+B) = \exp(A) \exp(B)$), 剩余项为:

$$\begin{aligned} Q_\tau(\tau) &\propto \exp(E_{Q_X(X)Q_W(W)} \{ \log p(Y, X, W, \tau) \}) \\ &\propto \exp \left((a-1) \log \tau - b\tau + \frac{NM}{2} \log \tau \right) \\ &\quad \times \exp \left(-\frac{\tau}{2} E_{Q_X(X)Q_W(W)} \left\{ \sum_n \sum_m (y_{nm} - w_m^T x_n)^2 \right\} \right) \end{aligned}$$

前文提到:

$$E_{p(a)} \{ f(a) + g(a) \} = E_{p(a)} \{ f(a) \} + E_{p(a)} \{ g(a) \}$$

对所有项取期望并分别相加。 y_{nm}^2 也是可观测数据, 因此:

$$E_{Q_{x_n}(x_n)Q_{w_m}(w_m)} \{ y_{nm}^2 \} = y_{nm}^2$$

253

254

因此, 只需考虑

$$\exp\left(-\frac{\tau}{2} \sum_{n,m} (y_{nm}^2 + \mathbf{E}_{Q_{x_n}(x_n)Q_{w_m}(w_m)}\{-2\mathbf{w}_m^T \mathbf{x}_n + \mathbf{x}_n^T \mathbf{w}_m \mathbf{w}_m^T \mathbf{x}_n\})\right)$$

第一眼看起来还是很难, 但是, 考虑下面的期望:

$$\mathbf{E}_{p(a)p(b)}\{f(a)f(b)\}$$

展开得到:

$$\begin{aligned}\mathbf{E}_{p(a)p(b)}\{f(a)f(b)\} &= \iint p(a)p(b)f(a)f(b)da db \\ &= \iint p(a)f(a)da p(b)f(b)db \\ &= \int \mathbf{E}_{p(a)}\{f(a)\}p(b)f(b) \\ &= \mathbf{E}_{p(a)}\{f(a)\}\mathbf{E}_{p(b)}\{f(b)\}\end{aligned}\quad (7-11)$$

使用这个结果, 可以衡量表达式中第一个参数:

$$\mathbf{E}_{Q_{x_n}(x_n)Q_{w_m}(w_m)}\{-2\mathbf{w}_m^T \mathbf{x}_n\} = -2\mathbf{E}_{Q_{x_n}(x_n)}\{\mathbf{x}_n\}^T \mathbf{E}_{Q_{w_m}(w_m)}\{\mathbf{w}_m\}$$

这是 \mathbf{x}_n 的 (均值) 期望值乘以 \mathbf{w}_m 的期望值。在继续下面的工作之前, 有必要先介绍一个更加有用的概念, 因为后面会有很多这样的表达形式。从现在起, 期望定义为以下形式:

$$\mathbf{E}_{Q_{\theta_l}(\theta_l)}\{f(\theta_l)\} = \langle f(\theta_l) \rangle$$

期望中的第一项写作:

$$\mathbf{E}_{Q_{x_n}(x_n)Q_{w_m}(w_m)}\{-2\mathbf{w}_m^T \mathbf{x}_n\} = -2\langle \mathbf{x}_n \rangle^T \langle \mathbf{w}_m \rangle$$

第二项需要一点小技巧, 不能把它写作 $f(\mathbf{x}_n)g(\mathbf{w}_n)$, 因此只能一次求期望

$$\begin{aligned}\mathbf{E}_{Q_{x_n}(x_n)Q_{w_m}(w_m)}\{\mathbf{x}_n^T \mathbf{w}_m \mathbf{w}_m^T \mathbf{x}_n\} &= \mathbf{E}_{Q_{x_n}(x_n)}\{\mathbf{x}_n^T \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \mathbf{x}_n\} \\ &= \langle \mathbf{x}_n^T \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \mathbf{x}_n \rangle\end{aligned}$$

把每一项放在一起, 有:

$$\begin{aligned}Q_\tau(\tau) &\propto \exp\left((a-1)\log \tau - b\tau + \frac{NM}{2}\log \tau\right. \\ &\quad \left.- \frac{\tau}{2} \sum_{n,m} (y_{nm}^2 - 2\langle \mathbf{w}_n \rangle^T \langle \mathbf{x}_n \rangle + \langle \mathbf{x}_n^T \langle \mathbf{w}_n \mathbf{w}_n^T \rangle \mathbf{x}_n) \right)\end{aligned}$$

255

可以写成

$$Q_\tau(\tau) \propto \tau^{e-1} \exp(-\tau f) \quad (7-12)$$

其中

$$\begin{aligned}e &= a + \frac{NM}{2} \\ f &= b + \frac{1}{2} \sum_{n,m} (y_{nm}^2 - 2\langle \mathbf{w}_n \rangle^T \langle \mathbf{x}_n \rangle + \langle \mathbf{x}_n^T \langle \mathbf{w}_n \mathbf{w}_n^T \rangle \mathbf{x}_n)\end{aligned}$$

式 (7-12) 的形式说明 $Q_\tau(\tau)$ 是参数为 e 、 f 的 γ 分布。如果怀疑这个结果, 可以对参数为 e 和 f 的 γ 密度函数求对数, 并去掉不依赖于 τ 的项——可得式 (7-12) 右侧的结果。

总之:

$$Q_\tau(\tau) = \Gamma(e, f)$$

现在, 得到 $Q_{x_n}(x_n)$ 和 $Q_{w_m}(w_m)$ ——得到这些形式就可以计算 e 和 f 所需要的期望值。

7.5.2 $Q_{x_n}(x_n)$

为了获得 $Q_{x_n}(x_n)$, 所需要的步骤很多, 且都与得到 $Q_\tau(\tau)$ 需要的步骤相同。开始, 我们从 $\log p(\mathbf{Y}, \mathbf{X}, \mathbf{W}, \tau)$ 中提取需要的全部项, 忽略包括 \mathbf{x}_n 的所有项:

$$Q_{x_n}(x_n) \propto \exp\left(E_{Q_w(w)Q_\tau(\tau)}\left\{\log p(x_n) + \sum_{m=1}^M p(y_{nm} | w_m, x_n, \tau)\right\}\right)$$

注意, 对于所有的 $l \neq n$ 期望也应该与所有的 $Q_{x_l}(x_l)$ 相关。然而, 在我们的表达式中不存在 x_l 项, 因此期望也就不存在了。在期望中展开这两项并且删除没有 x_n 的项, 我们有:

$$\begin{aligned} Q_{x_n}(x_n) &\propto \exp\left(E_{Q_w(w)Q_\tau(\tau)}\left\{-\frac{1}{2}x_n^T x_n - \frac{1}{2}\tau \sum_m (-2y_{nm}x_n^T w_m + x_n^T w_m w_m^T x_n)\right\}\right) \\ &\propto \exp\left(-\frac{1}{2}x_n^T x_n - \frac{1}{2}\langle\tau\rangle \sum_m (-2y_{nm}x_n^T \langle w_m \rangle + x_n^T \langle w_m w_m^T \rangle x_n)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(x_n^T \left[I_D + \langle\tau\rangle \sum_m \langle w_m w_m^T \rangle\right] x_n - 2\langle\tau\rangle x_n^T \sum_m y_{nm} \langle w_m \rangle\right)\right) \end{aligned}$$

期望中出现的线性和平方项告诉我们, 这是一个高斯分布:

$$Q_{x_n}(x_n) = \mathcal{N}(\mu_{x_n}, \Sigma_{x_n})$$

256

利用系数相等我们可以从表达式中得到 μ_{x_n} 和 Σ_{x_n} :

$$\begin{aligned} x_n^T \Sigma_{x_n}^{-1} x_n &\equiv x_n^T \left[I_D + \langle\tau\rangle \sum_m \langle w_m w_m^T \rangle \right] x_n \\ \Sigma_{x_n} &= \left[I_D + \langle\tau\rangle \sum_m \langle w_m w_m^T \rangle \right]^{-1} \end{aligned} \quad (7-13)$$

$$\begin{aligned} -2x_n^T \Sigma_{x_n}^{-1} \mu_{x_n} &\equiv -2\langle\tau\rangle x_n^T \sum_m y_{nm} \langle w_m \rangle \\ \mu_{x_n} &= \langle\tau\rangle \Sigma_{x_n} \sum_m y_{nm} \langle w_m \rangle \end{aligned} \quad (7-14)$$

注意, 协方差矩阵 Σ_{x_n} 不依赖于 n 。它只需要计算一次并可以应用于所有的 x_n 。

7.5.3 $Q_{w_m}(w_m)$

计算 $Q_{w_m}(w_m)$ 的方法在本质上与计算 $Q_{x_n}(x_n)$ 的方法是相同的。我们从删除所有不包含 w_n 的项开始:

$$Q_{w_m}(w_m) \propto \exp\left(E_{Q_{x_n}(x_n)Q_\tau(\tau)}\left\{\log p(w_m) + \sum_{n=1}^N p(y_{nm} | w_m, x_n, \tau)\right\}\right)$$

再次, 对于所有的 $l \neq m$ 关于 $Q_{w_l}(w_l)$ 的期望都不存在了。扩展, 注意 $w_m^T x_n = x_n^T w_m$:

$$\begin{aligned} Q_{w_m}(w_m) &\propto \exp\left(E_{Q_{x_n}(x_n)Q_\tau(\tau)}\left\{-\frac{1}{2}w_m^T w_m - \frac{1}{2}\tau \sum_n (-2y_{nm}w_m^T x_n + w_m^T x_n x_n^T w_m)\right\}\right) \\ &\propto \exp\left(-\frac{1}{2}w_m^T w_m - \frac{1}{2}\langle\tau\rangle \sum_n (-2y_{nm}w_m^T \langle x_n \rangle + w_m^T \langle x_n x_n^T \rangle w_m)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(w_m^T \left[I_D + \langle\tau\rangle \sum_n \langle x_n x_n^T \rangle\right] w_m - 2\langle\tau\rangle w_m^T \sum_n y_{nm} \langle x_n \rangle\right)\right) \end{aligned}$$

很明显, 这是另一个高斯分布:

$$Q_{w_m}(w_m) = \mathcal{N}(\mu_{w_m}, \Sigma_{w_m})$$

$$\Sigma_{w_m} = \left[I_D + \langle\tau\rangle \sum_n \langle x_n x_n^T \rangle \right]^{-1}$$

$$\mu_{w_m} = \langle\tau\rangle \Sigma_{w_m} \sum_n y_{nm} \langle x_n \rangle$$

与 Σ_{x_n} 一样, 协方差矩阵 Σ_{w_m} 不依赖于 m 并且对于所有的 w_m , Σ_{w_m} 只需要计算一次。

257

7.5.4 期望值要求

我们得到的近似后验 $Q_{x_n}(x_n)$ 、 $Q_{w_m}(w_m)$ 、 $Q_\tau(\tau)$ 的每一部分都依赖于与其他部分相关的期望 (例如, $\langle x_n \rangle$ 和 $\langle w_m w_m^T \rangle$)。由于所有的部分都是常见分布, 所以这些期望都是标准

的结果。 $Q_{x_n}(x_n)$ 和 $Q_{w_m}(w_m)$ 都服从高斯分布并且有:

$$\begin{aligned}\langle x_n \rangle &= \mu_{x_n} & \langle x_n x_n^T \rangle &= \Sigma_{x_n} + \mu_{x_n} \mu_{x_n}^T \\ \langle w_m \rangle &= \mu_{w_m} & \langle w_m w_m^T \rangle &= \Sigma_{w_m} + \mu_{w_m} \mu_{w_m}^T\end{aligned}$$

$Q_\tau(\tau)$ 是 γ 分布, 因此:

$$\langle \tau \rangle = \frac{e}{f}$$

我们想要的最后的期望是 $\langle x_n^T \langle w_m w_m^T \rangle x_n \rangle$ 。这属于 $\langle z^T A z \rangle$ 形式, 其中如果 $p(z) = \mathcal{N}(\mu, \Sigma)$ 则等同于:

$$\langle z^T A z \rangle = \text{Tr}(A \Sigma) + \mu^T A \mu$$

因此:

$$\langle x_n^T \langle w_m w_m^T \rangle x_n \rangle = \text{Tr}(\langle w_m w_m^T \rangle \Sigma_{x_n}) + \mu_{x_n}^T \langle w_m w_m^T \rangle \mu_{x_n}$$

7.5.5 算法

我们现在已经利用变分贝叶斯方法 (VB) 得到了获得近似后验 $Q(W, X, \tau)$ 的全部准备。首先我们必须初始化参数。我们将以初始化 $\langle \tau \rangle = a/b$ (期望优先值) 开始, 然后从分布 $\langle w_m \rangle$ 中抽样每一个 $\mathcal{N}(\theta, I_D)$, 并计算 $\langle w w^T \rangle = I_D + \langle w_m \rangle \langle w_m \rangle^T$ 。现在我们能计算 μ_{x_n} 和 Σ_{x_n} , 因此也能计算出 $\langle x_n \rangle$ 和 $\langle x_n x_n^T \rangle$ 。步骤如下:

- 1) 对于所有的 n , 计算 Σ_{x_n} 和 μ_{x_n} , 并且更新 $\langle x_n \rangle$ 和 $\langle x_n x_n^T \rangle$ 。
- 2) 对于所有的 m , 使用新产生的 $\langle x_n \rangle$ 和 $\langle x_n x_n^T \rangle$, 计算 μ_{w_m} 和 Σ_{w_m} 并更新 $\langle w_m \rangle$ 和 $\langle w_m w_m^T \rangle$ 。
- 3) 对于所有的 n 和 m 计算 $\langle x_m^T \langle w_m w_m^T \rangle x_n \rangle$ 。
- 4) 计算 e 和 f , 并更新 $\langle \tau \rangle$ 。
- 5) 如果不收敛, 则返回到 1)。

为了检查收敛性, 我们可以监测参数的变化情况也可以计算边界 $\mathcal{L}(\theta)$ (式 (7-6)), 它将递增至收敛, 然后保持不变。边界由下式给出:

$$\begin{aligned}\mathcal{L}(X, W, \tau) &= \int Q(X, W, \tau) \log \frac{p(Y, X, W, \tau)}{Q(X, W, \tau)} dQ(X, W, \tau) \\ &= \int Q(\cdot) \log p(\cdot) dQ(\cdot) - \int Q(\cdot) \log Q(\cdot) dQ(\cdot)\end{aligned}$$

利用独立性假设并注意两个表达式都是和 $Q(\cdot)$ 相关的期望, 我们可以进一步分解这两项:

$$\begin{aligned}\int Q(\cdot) \log p(\cdot) dQ(\cdot) &= E_{Q_\tau(\tau)} \{ \log p(\tau | a, b) \} \\ &+ \sum_{n=1}^N E_{Q_{x_n}(x_n)} \{ \log p(x_n) \} \\ &+ \sum_{m=1}^M E_{Q_{w_m}(w_m)} \{ \log p(w_m) \} \\ &+ \sum_{n=1}^N \sum_{m=1}^M E_{Q_{x_n}(x_n) Q_{w_m}(w_m) Q_\tau(\tau)} \{ \log p(y_{nm} | x_n, w_m, \tau) \}\end{aligned}$$

和:

$$\begin{aligned}\int Q(\cdot) \log Q(\cdot) dQ(\cdot) &= E_{Q_\tau(\tau)} \{ \log Q_\tau(\tau) \} \\ &+ \sum_{n=1}^N E_{Q_{x_n}(x_n)} \{ \log Q_{x_n}(x_n) \} \\ &+ \sum_{m=1}^M E_{Q_{w_m}(w_m)} \{ \log Q_{w_m}(w_m) \}\end{aligned}$$

这些单独项给出边界（按照上面的顺序每一行对应一个期望）留给读者证明（详见练习 EX 7.2）:

$$\begin{aligned}
 \mathcal{L}(\mathbf{X}, \mathbf{W}, \tau) = & a \log b + (a-1) \langle \log \tau \rangle - b \langle \tau \rangle - \log \Gamma(a) \\
 & - \frac{ND}{2} \log 2\pi - \frac{1}{2} \sum_n (\text{Tr}(\mathbf{\Sigma}_{x_n}) + \boldsymbol{\mu}_{x_n}^T \boldsymbol{\mu}_{x_n}) \\
 & - \frac{MD}{2} \log 2\pi - \frac{1}{2} \sum_m (\text{Tr}(\mathbf{\Sigma}_{w_m}) + \boldsymbol{\mu}_{w_m}^T \boldsymbol{\mu}_{w_m}) \\
 & - \frac{NM}{2} \log 2\pi + \frac{NM}{2} \langle \log \tau \rangle - \frac{1}{2} \langle \tau \rangle \sum_{n,m} \langle (y_{nm} - \mathbf{w}_m^T \mathbf{x}_n)^2 \rangle \\
 & - (e \log f + (e-1) \langle \log \tau \rangle - f \langle \tau \rangle - \log \Gamma(e)) \\
 & - \left(-\frac{ND}{2} \log 2\pi - \frac{ND}{2} - \frac{1}{2} \sum_n \log |\mathbf{\Sigma}_{x_n}| \right) \\
 & - \left(-\frac{MD}{2} \log 2\pi - \frac{MD}{2} - \frac{1}{2} \sum_m \log |\mathbf{\Sigma}_{w_m}| \right)
 \end{aligned}$$

其中,

$$\langle (y_{nm} - \mathbf{w}_m^T \mathbf{x}_n)^2 \rangle = y_{nm}^2 - 2y_{nm} \langle \mathbf{x}_n \rangle^T \langle \mathbf{w}_m \rangle + \langle \mathbf{x}_n^T \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \mathbf{x}_n \rangle$$

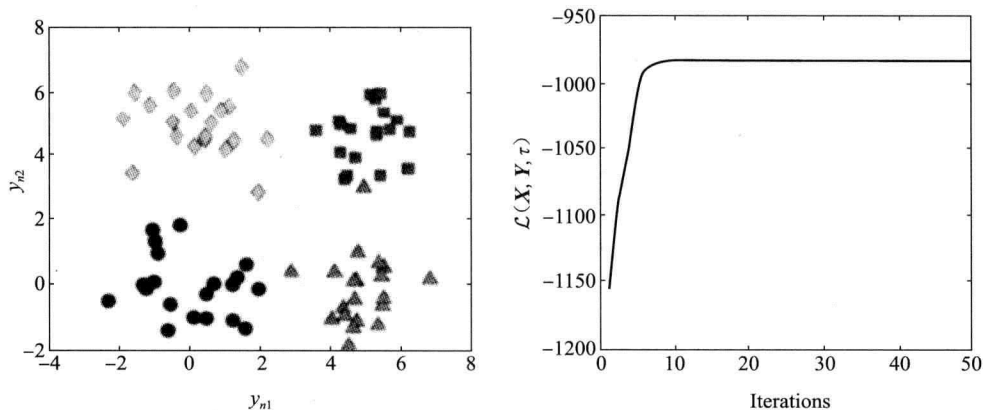
所有项都已经计算出来。在边界上我们以前没有见过的唯一一项是 $\langle \log \tau \rangle$ 。我们不得不估计这一项，我们可以通过采样做到这点。如果我们抽取 S 个样本 τ^1, \dots, τ^S ，按照如下方法给出估计：

259

$$\langle \log \tau \rangle \approx \frac{1}{S} \sum_{s=1}^S \log \tau^s$$

7.5.6 例子

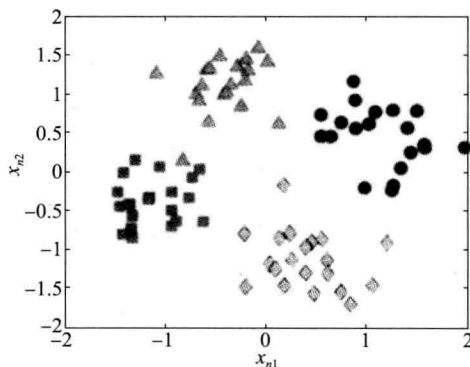
图 7-6a 展示了一个数据集，这个数据集的生成方法和图 7-4（MATLAB 脚本：ppcae-example.m）描述的例子是相同的。在两维中有很清晰的聚类结构。此外，额外的 5 维被加入 ($y_{nm} \sim \mathcal{N}(0, 1)$)。边界演化过程 $\mathcal{L}(\mathbf{X}, \mathbf{W}, \tau)$ 作为算法的执行过程 ($D=2$) 可以在图 7-6b 中看到。边界单调递增直到收敛，这仅仅需要很少的迭代次数。在图 7-6c 中我们能看到隐变量的后验方法。这和标准的 PCA 投影（图 7-4c）有很多相似之处。明显聚类结构是在隐变量空间中形成的。



a) 对象 y_n 的前两维数据

b) 下边界 $\mathcal{L}(\mathbf{X}, \mathbf{Y}, \tau)$ 的变化过程

图 7-6 PCA 例子的合成概率



c) 隐变量的后验均值

图 7-6 (续)

7.6 缺失值

我们把表示方法变成概率形式的目的之一就是有利于处理缺失值。在前面章节定义的模型中，解决这个问题的最早方法是仅仅为观测数据定义一个模型。为了这个目的，我们引进了一套新的二元变量 z_{nm} ，当我们观察对象 n 的特征 m 时，变量为 1；否则为 0。对于所有的 z_{nm} ，得到矩阵 \mathbf{Z} ，我们有：

$$p(\mathbf{Y}, \mathbf{X}, \mathbf{W}, \tau | \mathbf{Z}) = p(\tau | a, b) \left[\prod_{m=1}^M p(\mathbf{w}_m) \right] \left[\prod_{n=1}^N p(\mathbf{x}_n) \prod_{m=1}^M p(y_{nm} | \mathbf{w}_m, \mathbf{x}_n, \tau)^{z_{nm}} \right]$$

$$\log p(\mathbf{Y}, \mathbf{X}, \mathbf{W}, \tau | \mathbf{Z}) = \log p(\tau | a, b) + \sum_{m=1}^M \log p(\mathbf{w}_m) + \sum_{n=1}^N \log p(\mathbf{x}_n) \quad (7-15)$$

$$+ \sum_{n=1}^N \sum_{m=1}^M z_{nm} \log p(y_{nm} | \mathbf{w}_m, \mathbf{x}_n, \tau)$$

二元变量的作用是充当开关，只提供我们观察的数据项。注意，上述这些如何受限于 \mathbf{Z} 。根据前面章节的详细步骤推导出必要的变量分布留给读者作为练习（详见练习 EX 7.3）。这些是：

$$Q_{\mathbf{x}_n}(\mathbf{x}_n) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}_n}, \boldsymbol{\Sigma}_{\mathbf{x}_n})$$

$$\boldsymbol{\Sigma}_{\mathbf{x}_n} = \left[\mathbf{I}_D + \langle \tau \rangle \sum_m z_{nm} \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \right]^{-1}$$

$$\boldsymbol{\mu}_{\mathbf{x}_n} = \langle \tau \rangle \boldsymbol{\Sigma}_{\mathbf{x}_n} \sum_m z_{nm} y_{nm} \langle \mathbf{w}_m \rangle$$

$$Q_{\mathbf{w}_m}(\mathbf{w}_m) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{w}_m}, \boldsymbol{\Sigma}_{\mathbf{w}_m})$$

$$\boldsymbol{\Sigma}_{\mathbf{w}_m} = \left[\mathbf{I}_D + \langle \tau \rangle \sum_n z_{nm} \langle \mathbf{x}_n \mathbf{x}_n^T \rangle \right]^{-1}$$

$$\boldsymbol{\mu}_{\mathbf{w}_m} = \langle \tau \rangle \boldsymbol{\Sigma}_{\mathbf{w}_m} \sum_n z_{nm} y_{nm} \langle \mathbf{x}_n \rangle$$

$$Q_{\tau}(\tau) = \Gamma(e, f)$$

$$e = a + \frac{1}{2} \sum_{n,m} z_{nm}$$

$$f = b + \frac{1}{2} \sum_{n,m} z_{nm} (y_{nm}^2 - 2 \langle \mathbf{w}_m \rangle^T \langle \mathbf{x}_n \rangle + \langle \mathbf{x}_n^T \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \mathbf{x}_n \rangle)$$

在前面的章节中，我们已经提到过 $\boldsymbol{\Sigma}_{\mathbf{x}_n}$ 和 $\boldsymbol{\Sigma}_{\mathbf{w}_m}$ 的方程分别不依赖于 n 和 m ，相反，不需要对每个 n 和 m 都进行计算。由于在两个表达式中都存在 z_{nm} ，所以不会再有这种情况，并且对

于每一个 \mathbf{x}_n 也不用再计算 $\Sigma_{\mathbf{x}_n}$ (对每个 \mathbf{w}_m 也同样不用再计算 $\Sigma_{\mathbf{w}_m}$)。 \mathbf{x}_n 和 \mathbf{w}_m 都是 D 维的, 所以对大的 N 、 M 和 D , 这将是很大的计算量。在 7.4.1 节中, 当定义近似后验组分时, 我们提到了提出额外的独立假设的可能性。特别地:

$$Q_{\mathbf{x}_n}(\mathbf{x}_n) = \prod_{d=1}^D Q_{x_{nd}}(x_{nd})$$

这将使得我们必须使用标量方差而不是 $D \times D$ 的协方差矩阵 $\Sigma_{\mathbf{x}_n}$ 。这将很大程度上降低计算量, 但是以更坏的后验近似程度为代价的。

261

图 7-7 给出了带有缺失值的概率性 PCA 模型 (MATLAB 脚本: `ppcamvexample.m`)。图 7-7b 给出了变元后验 $Q_{\mathbf{x}_n}(\mathbf{x}_n)$ 的后验均值。数据与前面例子中的数据相同 (图 7-7a 中给出了数据的前两个维度的簇结构, 剩下 5 个维中的噪声), 其中每一个 y_{nm} 的值以 0.05 的概率变化。正如我们期望的那样, 移动数据的影响是聚类结构不那么明显了。单独的协方差矩阵使这个影响更容易表现出来。在图 7-7c 中, 我们把三个圆类对象看做椭圆, 使协方差矩阵更加具体化。椭圆告诉我们模型把不确定程度归结于隐变量的值。在这两维中对对象 1 没有缺失值, 这些值可以影响簇结构并与同类中的其他对象具有相似的特征。对象 2 关于 y_{n2} 的值是缺失的, 也就是那些决定样本是属于圆类还是钻石 (见图 7-7a) 类的信息。在图 7-7c 中这表现在它的均值和方差上了——这个模型使得样本位于不同团的中间, 但是也存在它位二者中的任何一个内的可能性。3 缺失 y_{n1} 和 y_{n2} ——全部非噪声特征。模型使它放到离原点很近 (记得前验 $p(\mathbf{x}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$), 但具有很高的不确定性——可以属于任何组。

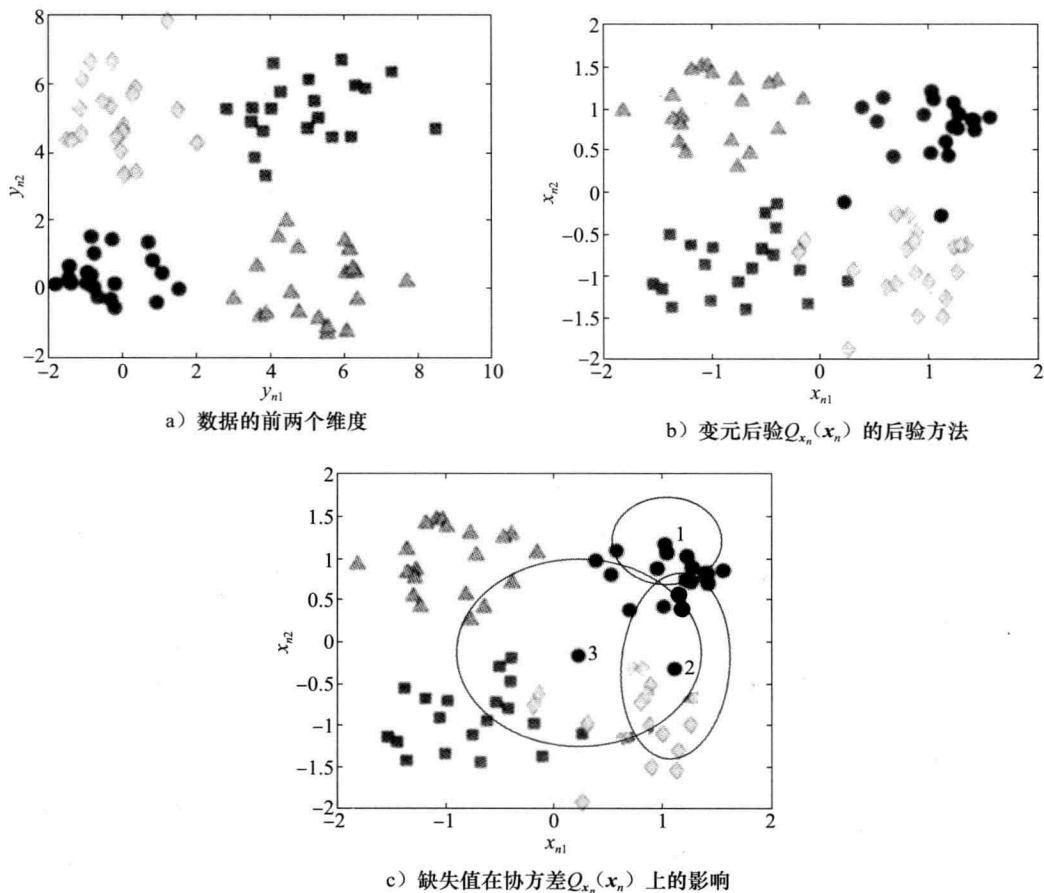


图 7-7 有缺失值的变分贝叶斯 PPCA 模型。数据是与图 7-6 中的一样, 其中每一个 y_{nm} 的值以 0.05 的概率变化

显然图 7-7c 中给出的协方差信息是模型给出的很有用的输出信息。知道关于隐变量空间中的对象 \mathbf{x}_n 位置是否有很高的不确定性是很重要的。换句话说, 如果我们仅看图 7-7c, 我们可能推断出, 我们不应该得出任何关于对象 3 位置的严重结论——因为缺失值导致协方差很高。在 7.7.2 节中, 我们将看到一个协方差信息很有用的有趣例子。

7.6.1 缺失值作为隐变量

在前面的章节中, 我们已经看到 VB 框架如何允许我们解决缺失值的问题——我们仅仅包含了模型中观察到的值。当一些缺失值通过单独的协方差矩阵 $\Sigma_{\mathbf{x}_n}$ 自然地处理时, 增长的不确定性也出现了。具有额外计算负担的单个协方差矩阵, 有些情况下, 这些负担过高。作为一个选择, 可以考虑把缺失值作为额外的隐变量。引进上标 h 和 o 分别表示隐藏的和可观测的。这相当于下面的联合对数似然:

$$\begin{aligned} \log p(\mathbf{Y}, \mathbf{X}, \mathbf{W}, \tau | \mathbf{Z}) = & \log p(\tau | a, b) + \sum_{m=1}^M \log p(\mathbf{w}_m) + \sum_{n=1}^N \log p(\mathbf{x}_n) \\ & + \sum_{n=1}^N \sum_{m=1}^M z_{nm} \log p(y_{nm}^o | \mathbf{w}_m, \mathbf{x}_n, \tau) \\ & + \sum_{n=1}^N \sum_{m=1}^M (1 - z_{nm}) \log p(y_{nm}^h | \mathbf{w}_m, \mathbf{x}_n, \tau). \end{aligned}$$

此外, 我们需要另一组变分后验: $Q_{y_{nm}^h}(y_{nm}^h)$ 。在此我们将忽略 VB 算法的推导, 但是会说明重要的结果。首先, 额外的变分后验:

$$Q_{y_{nm}^h}(y_{nm}^h) = \mathcal{N}(\langle \mathbf{w}_m \rangle^T \langle \mathbf{x}_n \rangle, \langle \tau \rangle^{-1})$$

所以, $\langle y_{nm}^h \rangle = \langle \mathbf{w}_m \rangle^T \langle \mathbf{x}_n \rangle$ 。 $Q_{\mathbf{x}_n}(\mathbf{x}_n)$ 是通过一个有下面参数的高斯分布给出的:

$$\begin{aligned} \Sigma_{\mathbf{x}_n} &= \left[\mathbf{I}_D + \langle \tau \rangle \sum_m z_{nm} \langle \mathbf{w}_m \mathbf{w}_m^T \rangle + \langle \tau \rangle \sum_m (1 - z_{nm}) \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \right]^{-1} \\ &= \left[\mathbf{I}_D + \langle \tau \rangle \sum_m \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \right]^{-1} \\ \mu_{\mathbf{x}_n} &= \langle \tau \rangle \Sigma_{\mathbf{x}_n} \sum_m (z_{nm} y_{nm}^o + (1 - z_{nm}) \langle y_{nm}^h \rangle) \langle \mathbf{w}_m \rangle \\ &= \langle \tau \rangle \Sigma_{\mathbf{x}_n} \sum_m y_{nm}^* \langle \mathbf{w}_m \rangle \end{aligned} \quad (7-16)$$

其中 y_{nm}^* 是一个具有元素 y_{nm}^o 和 $\langle y_{nm}^h \rangle$ 的向量, 它依赖于是否有特别的参数被观测到。 $Q_{\mathbf{w}_m}(\mathbf{w}_m)$ 和 $Q_{\tau}(\tau)$ 很相似。

从式 (7-16) 中可以清楚地看到 $Q_{\mathbf{x}_n}(\mathbf{x}_n)$ 的方差不再依赖 n , 因此不再对每个对象都需要一个特定的协方差矩阵。事实上, 可以观察到 VB 算法的结果与原始 VB PCA 的结果一样, 在这个算法中我们在每一个缺失值位置插入一些模型中期望的 y_{nm} (例如 $\langle \mathbf{w}_m \rangle^T \langle \mathbf{x}_n \rangle$) 处缺失的值。这在很大程度上降低了计算量, 但我们丢掉了目标的协方差矩阵 $\Sigma_{\mathbf{x}_n}$ 。在隐变量空间中, 所有的目标有相同的协方差, 不考虑有多少观测或者丢失的值, 因为没观测数据的期望值 $\langle y_{nm}^h \rangle$ 与实际数据具有同样的影响。如果缺失值很少, 那么这可能不是问题。如果有许多缺失值, 那么应该避免。

7.6.2 预测缺失值

考虑将缺失值作为隐变量的好处之一是自动地把错误归咎于缺失值。然而, 我们仍然能够使用原始的缺失值模型做到这一点。特别地, 与变分后验相关的 y_{nm}^h 的期望值是:

$$\begin{aligned} E_{Q(\cdot)} \{ y_{nm}^h \} &= E_{Q(\cdot)} \{ \mathbf{w}_m^T \mathbf{x}_n + \epsilon \} \\ &= \langle \mathbf{w}_m \rangle^T \langle \mathbf{x}_n \rangle \end{aligned}$$

其中 $\epsilon \mathcal{N}(0, \tau^{-1})$ 。预测值的方差为:

$$\begin{aligned}\text{var}\{y_{nm}^h\} &= \mathbf{E}_{Q(\cdot)}\{(\mathbf{y}_{nm}^h)^2\} - \mathbf{E}_{Q(\cdot)}\{\mathbf{y}_{nm}^h\}^2 \\ &= \langle \mathbf{x}_n^T \langle \mathbf{w}_m \mathbf{w}_m^T \mathbf{x}_n \rangle + \langle \tau \rangle^{-1} - \langle \mathbf{x}_n \rangle^T \langle \mathbf{w}_m \rangle \langle \mathbf{w}_m \rangle^T \langle \mathbf{x}_n \rangle\end{aligned}$$

作为一个例子, 考虑图 7-7c 中的对象 2——用这些表达式, 缺失值 y_{n2}^h 有均值 0.5839 和方差 5.4070。

7.7 非实值数据

为了处理非实值数据是 7.2.2 节讨论过的为了实现概率性表示的第二个目的。在我们介绍 VB PPCA 时, 我们使用了高斯似然。使用相同的步骤, 我们可以使用 VB 推导出具有替代可能性的近似 PCA 模型。由非实值数据和缺失值组成的一个有趣的数据集来自 Public whip (<http://www.publicwhip.org.uk>) 的可用的英国议会成员选举投票历史。英国的议会成员选举是在普通的选举中选举国会成员。一届国会大约持续 4~5 年, 在这段时间里大约举行 1000 次选举。每张选票由二值选择 (议会成员或者完全支持或者完全反对议员的提议) 组成。议会成员不一定要投票, 他们可以选择弃权, 或者在选举当天不出席。所以, 这些数据既是非实值的也包含了一些缺失值。

正如我们在第 4 章中看到的一样, 二值数据通常都伴随着分析性问题。不必重温那一章的内容, 我们现在将展示一个基于引进一个辅助变量 (隐藏变量) 的选择性方法。表明 PPCA 模型不是解决这个问题的唯一途径并不是我们的目的, 但这是一个处理二值似然更一般技术的很好实例。我们建议读者可以阅读本章末尾的选择性概率二值 PCA 算法。

7.7.1 概率 PPCA

我们观察了 N 个议会成员的 M 张选票。对每一张选票, 假设没有缺失值 (例如, $z_{nm}=1$), 我们观察 $y_{nm}=\pm 1$ 。如以前一样, 我们将假设有一些 D 维的未被观察到的由一组向量 \mathbf{w}_m 映射的隐变量 \mathbf{x}_n 。在前面的例子中, 我们对 $p(y_{nm}|\mathbf{w}_m, \mathbf{x}_n)$ 使用了高斯似然。为了模拟二值的议会成员数据, 我们将使用概率似然作为替代。概率函数 (也称为正态条件密度函数) 定义如下:

$$\phi(z) = \int_{-\infty}^z \exp\left\{-\frac{1}{2}x^2\right\} dx$$

并把一个实值的变量 z 转换为 $0\sim 1$ (类似于第 4 章中用于逻辑回归的 sigmoid 函数)。特别地, 我们将定义:

$$P(y_{nm}=1|\mathbf{w}_m, \mathbf{x}_n) = \phi(\mathbf{w}_m^T \mathbf{x}_n) \quad (7-17)$$

和

$$P(y_{nm}=-1|\mathbf{w}_m, \mathbf{x}_n) = 1 - P(y_{nm}=1|\mathbf{w}_m, \mathbf{x}_n)$$

不幸的是, 如果我们试着阐述变分后验 $Q_{\mathbf{x}_n}(\mathbf{x}_n)$, 我们将发现它们不属于任何可识别的形式。此时, 我们使用一个略微奇怪的技巧。我们从引入一组新的 (实值) 变量 q_{nm} 开始:

$$p(q_{nm}|\mathbf{w}_m, \mathbf{x}_n) = \mathcal{N}(\mathbf{w}_m^T \mathbf{x}_n, 1)$$

我们通过下面的似然函数, 连接观测到的数据 y_{nm} :

$$P(y_{nm}=1|q_{nm}) = \delta(q_{nm} > 0)$$

和

$$P(y_{nm}=-1|q_{nm}) = \delta(q_{nm} < 0)$$

为了证明选择的正确性, 考虑 q_{nm} 和 y_{nm} 的联合分布:

$$p(y_{nm}=1, q_{nm}|\mathbf{w}_m, \mathbf{x}_n) = P(y_{nm}=1|q_{nm})p(q_{nm}|\mathbf{w}_m, \mathbf{x}_n)$$

选择 $P(y_{nm}=1|q_{nm})=\delta(q_{nm}>0)$ 意味着如果我们忽略 q_{nm} , 我们将重新回到原始的概率似然 (式 (7-17)):

$$\begin{aligned}
 P(y_{nm}=1|\mathbf{w}_m, \mathbf{x}_n) &= \int p(y_{nm}=1, q_{nm}|\mathbf{w}_m, \mathbf{x}_n) dq_{nm} \\
 &= \int P(y_{nm}=1|q_{nm}) p(q_{nm}|\mathbf{w}_m, \mathbf{x}_n) dq_{nm} \\
 &= \int_{-\infty}^{\infty} \delta(q_{nm}>0) \mathcal{N}(\mathbf{w}_m^T \mathbf{x}_n, 1) dq_{nm} \\
 &= \int_0^{\infty} \mathcal{N}(\mathbf{w}_m^T \mathbf{x}_n, 1) dq_{nm} \\
 &= \int_{-\mathbf{w}_m^T \mathbf{x}_n}^{\infty} \mathcal{N}(0, 1) dq_{nm} \\
 &= \int_{-\infty}^{\mathbf{w}_m^T \mathbf{x}_n} \mathcal{N}(0, 1) dq_{nm} = \phi(\mathbf{w}_m^T \mathbf{x}_n)
 \end{aligned}$$

265

这表明我们可以把概率似然作为具有附加参数 q_{nm} 的模型的结果, 并且已经被整理过。这表明如果这个参数 (我们把 q_{nm} 看做是一个隐变量, 并推断它的值) 被留在 VB 算法中, 那么将变得十分简单, 即使有一个额外的 $N \times M$ 阶的参数。图 7-8 描绘了模型表示的含义。

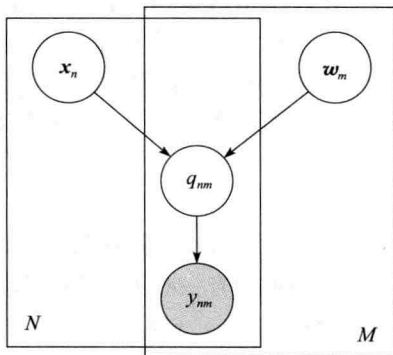


图 7-8 概率 PCA 模型的图形表示

把所有的 q_{nm} 整理为一个 $N \times M$ 的矩阵 \mathbf{Q} , 下面是 VB 算法的开始点:

$$\begin{aligned}
 \log p(\mathbf{Y}, \mathbf{X}, \mathbf{W}, \mathbf{Q}) &= \log \left[\prod_{m=1}^M p(\mathbf{w}_m) \right] \left[\prod_{n=1}^N p(\mathbf{x}_n) \right] \\
 &\quad \times \left[\prod_{n=1}^N \prod_{m=1}^M p(y_{nm}|q_{nm})^{z_{nm}} p(q_{nm}|\mathbf{w}_m, \mathbf{x}_n)^{z_{nm}} \right] \\
 &= \sum_{m=1}^M \log p(\mathbf{w}_m) + \sum_{n=1}^N \log p(\mathbf{x}_n) \\
 &\quad + \sum_{n=1}^N \sum_{m=1}^M z_{nm} [\log p(y_{nm}|q_{nm}) + \log p(q_{nm}|\mathbf{w}_m, \mathbf{x}_n)]
 \end{aligned}$$

对于变分近似, 与以前一样, 我们需要 $Q_{\mathbf{x}_n}(\mathbf{x}_n)$ 和 $Q_{\mathbf{w}_m}(\mathbf{w}_m)$, 还有 $Q_{q_{nm}}(q_{nm})$ 。把所有包含 \mathbf{x}_n 和 \mathbf{w}_m 的项聚到一起, 我们发现在实值模型中它们与这些项是一致的, 除了用 q_{nm} 替换 y_{nm} , 并且方差是 1 而不是 τ^{-1} 。所以我们已经知道变分分布分别是什么:

$$\begin{aligned}
 Q_{\mathbf{x}_n}(\mathbf{x}_n) &= \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}_n}, \boldsymbol{\Sigma}_{\mathbf{x}_n}) \\
 \boldsymbol{\Sigma}_{\mathbf{x}_n} &= \left[\mathbf{I}_D + \sum_m z_{nm} \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \right]^{-1} \\
 \boldsymbol{\mu}_{\mathbf{x}_n} &= \boldsymbol{\Sigma}_{\mathbf{x}_n} \sum_m z_{nm} \langle q_{nm} \rangle \langle \mathbf{w}_m \rangle
 \end{aligned}$$

266

$$Q_{w_m}(w_m) = \mathcal{N}(\mu_{w_m}, \Sigma_{w_m})$$

$$\Sigma_{w_m} = \left[I_D + \sum_m z_{nm} \langle \mathbf{x}_n \mathbf{x}_n^T \rangle \right]^{-1}$$

$$\mu_{w_m} = \Sigma_{w_m} \sum_m z_{nm} \langle q_{nm} \rangle \langle \mathbf{x}_n \rangle$$

对于 $Q_{q_{nm}}(q_{nm})$ 我们需要做一些工作。记得式 (7-9)，我们知道 $Q_{q_{nm}}(q_{nm})$ 将以如下的形式给出：

$$Q_{q_{nm}}(q_{nm}) \propto \exp(\mathbf{E}_{Q_{\mathbf{x}_n}(\mathbf{x}_n) Q_{w_m}(w_m)} \{ \log p(y_{nm} | q_{nm}) + \log p(q_{nm} | w_m, \mathbf{x}_n) \})$$

孤立项只包含 q_{nm} ，我们有：

$$Q_{q_{nm}}(q_{nm}) \propto p(y_{nm} | q_{nm}) \exp \left\{ -\frac{1}{2} (q_{nm}^2 - 2q_{nm} \langle w_m \rangle^T \langle \mathbf{x}_n \rangle) \right\}$$

这是 $p(y_{nm} | q_{nm})$ 乘以高斯的形式：

$$Q_{q_{nm}}(q_{nm}) \propto p(y_{nm} | q_{nm}) \mathcal{N}(\langle w_m \rangle^T \langle \mathbf{x}_n \rangle, 1)$$

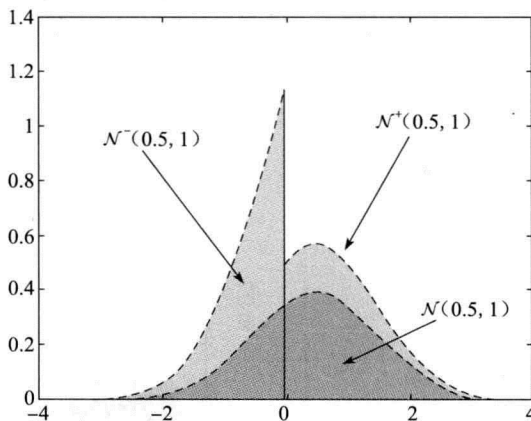
到此，我们将假设 $y_{nm} = 1$ 。因此，我们有：

$$Q_{q_{nm}}(q_{nm}) \propto \delta(q_{nm} > 0) \mathcal{N}(\langle w_m \rangle^T \langle \mathbf{x}_n \rangle, 1)$$

$$= \mathcal{N}^+(\langle w_m \rangle^T \langle \mathbf{x}_n \rangle, 1)$$

其中 $\mathcal{N}^+(\cdot)$ 用来表示高斯的截断（参见注解 7.3）以至于 q_{nm} 必须是正的。如果 $y_{nm} = -1$ 我们将以 $\mathcal{N}^-(\langle w_m \rangle^T \langle \mathbf{x}_n \rangle, 1)$ 的形式结束——一个具有相同均值截断的高斯分布，以至于 q_{nm} 必须是负的。

注解 7.3 (截断高斯密度)：截断高斯密度是在随机变量上有额外限制的高斯密度。我们只对划分在原点上或下的高斯密度感兴趣。



上图给出了标准的高斯密度（均值为 0.5、方差为 1），以及正相关和反相关的截断密度。截断密度和标准密度有相同的形状，但两个都较高。这是因为它们必须在减少的区域趋于 1。从截断高斯分布抽样是非常简单的——一个人仅能从未截断的密度抽样，并丢弃那些不满足必要限制的样本。正相关和反相关的截断高斯期望数据在下式给出：

$$p(x) = \mathcal{N}^+(\mu, \sigma^2), \quad \langle x \rangle = \mu + \frac{\mathcal{N}_{\mu/\sigma}(0, 1)}{1 - \phi(-\mu/\sigma)}$$

$$p(x) = \mathcal{N}^-(\mu, \sigma^2), \quad \langle x \rangle = \mu - \frac{\mathcal{N}_{\mu/\sigma}(0, 1)}{\phi(-\mu/\sigma)}$$

其中， $\mathcal{N}_a(0, 1)$ 是评估在 a 的标准高斯概率密度函数， $\phi(a)$ 是评估在 a 的标准正态条件密度函数。

为了计算 $Q_{x_n}(\mathbf{x}_n)$ 和 $Q_{w_m}(\mathbf{w}_m)$ 需要 $\langle q_{nm} \rangle$ 。这是作为截断高斯正相关或者反相关（依赖于 y_{nm} 的值）的期望值。计算它的通用表达式在注解 7.3 中给出。定义 $\mu_{nm} = \langle \mathbf{w}_m \rangle^T \langle \mathbf{x}_n \rangle$ 和 $\sigma = 1$ ，这些是：

$$y_{nm} = 1: \quad \langle q_{nm} \rangle = \mu_{nm} + \frac{\mathcal{N}_{\mu_{nm}}(0, 1)}{1 - \phi(-\mu_{nm})}$$

$$y_{nm} = -1: \quad \langle q_{nm} \rangle = \mu_{nm} - \frac{\mathcal{N}_{\mu_{nm}}(0, 1)}{\phi(-\mu_{nm})}$$

完成这个表达式需要 VB 算法。在 7.7.2 节中我们将给出一个使用该算法的例子。

7.7.2 议会数据可视化

研究这个模型的目的是议会成员的投票数据。我们将考察 2005—2010 年英国议会成员的投票数据。为了说明用一个合适的近似以及处理缺失值的明显优势，我们将用我们能使用的可视化数据的最简单的方法，即标准的主成分，来比较这个模型。此时我们用 0 表示缺失值（例如，一个值代表既可能是赞成票也可能是反对票， ± 1 ），并且对于数据不是真实值不做特殊考虑。

考虑到问题的复杂性情况，这个数据集由 657 个议会成员的 1288 次投票记录组成。每名议会成员的票数平均值是 853（66%），最积极的成员投了 1237 次票（96%），最不积极的成员投了 20 次（1.6%）。

图 7-9a 给出了在这个数据上使用标准 PCA 的结果。在隐空间中呈现出明显的聚类结构。在图 7-9b 中我们用他们所属的政党标注出议会成员，并且清楚地表明聚类结构符合 3 个主要的政党（劳动党、保守党、自由民主党）的情况。聚类结构表现出的情况不足为奇，因为议会成员投票通常是以政党情况分布。然而，在前两个主成分中很清楚地表现出这种情况是很鼓舞士气的。

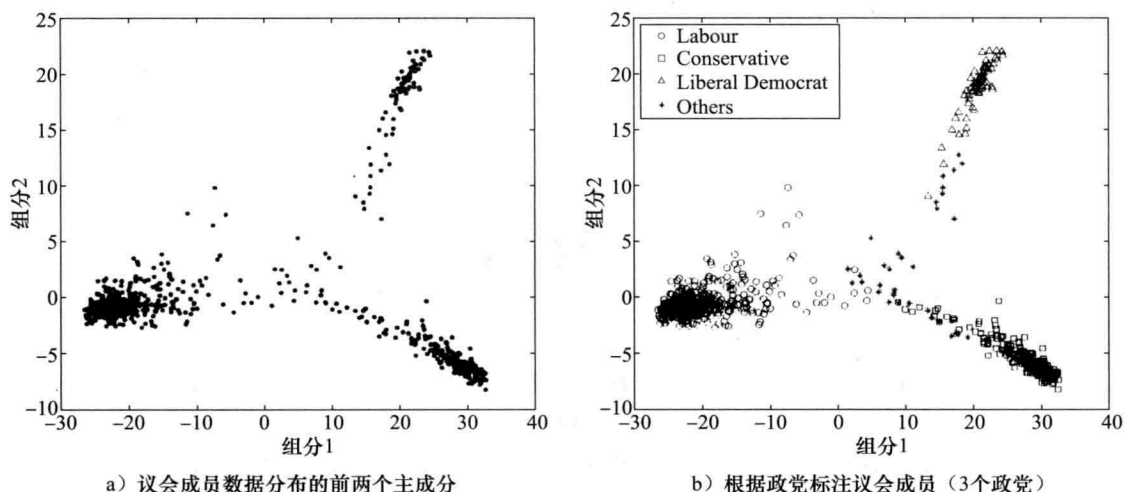


图 7-9 2005 年的议会成员投票数据的标准主成分，每一个点对应一个议会成员

有些议会成员似乎也被拉到了原点。这可以通过叛逆性进行解释，这些议会成员不赞成以前的政党路线。然而，不幸的是，这些议会成员通常只是简单地不投票。为了说明这点，如图 7-10 所示，我们能在 PCA 图中画出距离原点的投票数。很明显，大量的缺失值对分析这个问题没有什么帮助——隐变量空间中的位置是政策偏好和出席率的函数。

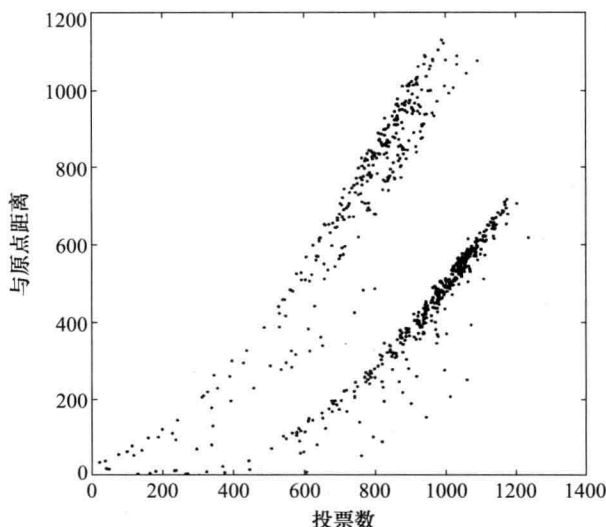


图 7-10 与 (PCA 图的) 原点距离相对的投票数

图 7-11a 给出了利用 VB 二值 PCA 算法得到的结果 (MATLAB 脚本: mpvis.m)。此图又一次清晰地展示了 $\langle \mathbf{x}_n \rangle$ 和聚类结构。在图 7-11b 中我们能看到通过政党标注的议会成员。聚类结构又一次与不同的政政策一致。因为我们在正确地模拟缺失值, 所以我们不再得到趋于原点的数据。为了说明它, 图 7-12 展示了远离原点的投票数目——在图 7-10 中呈现出的非常清晰的关系不再显著。图 7-11a 表现出的差异展示了政策趋势和缺席趋势。

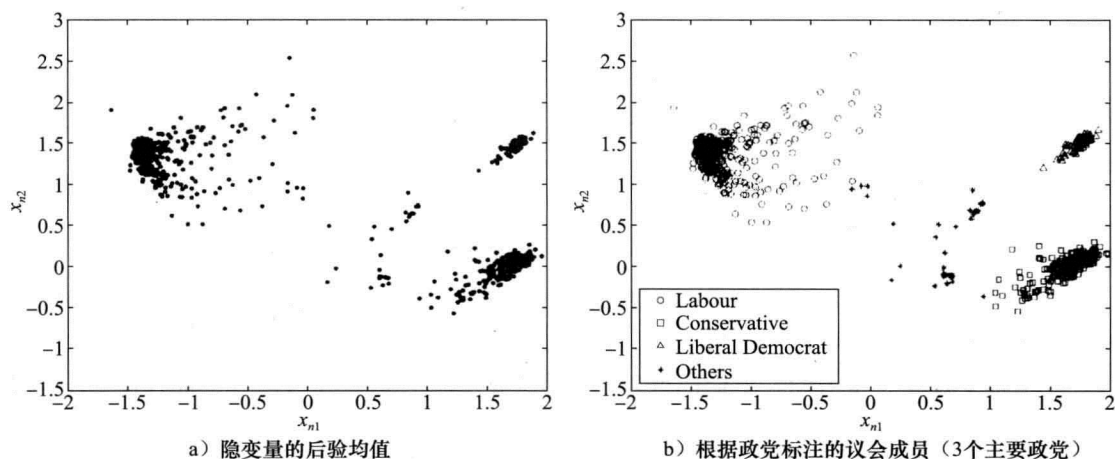


图 7-11 2005 年的议会成员投票数据的概率性二值主成分可视化, 每一个点对应一个成员

通过考虑议会中一些更小的党可以更好地说明这些。图 7-13 突出了 4 个小的政党的位置——民主工会党 (DUP)、威尔士党 (PC)、苏格兰国家党 (SNP) 和社会民主劳动党 (SDLP)。在传统的 PCA 分析中, 似乎 DUP 成员的票在保守党成员 (见图 7-9b) 的聚类内, 并且在更小的程度上, PC 和 SNP 成员的票趋向于自由民主党的选择。然而, 比较二值 PCA 算法的输出, 我们能够看到 DUP 清晰地形成自己的聚类, 远离保守党, 同时 SNP 和 PC 的成员形成了他们自己的一个很紧密的聚类。看起来好像原始 PCA 中的这些组的位置受有缺失值的较差模型影响很严重。

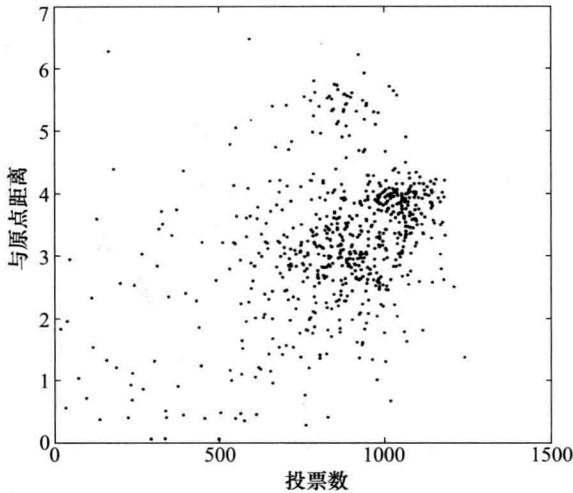
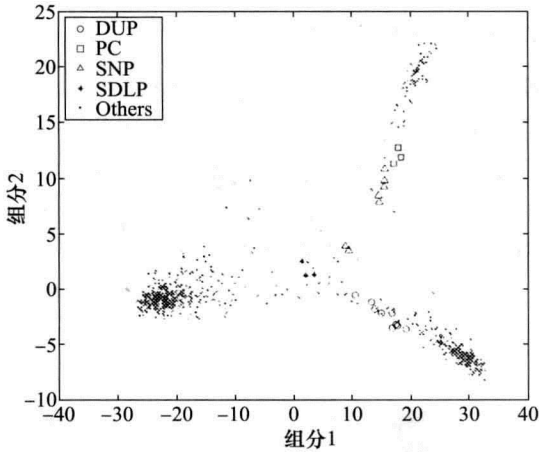
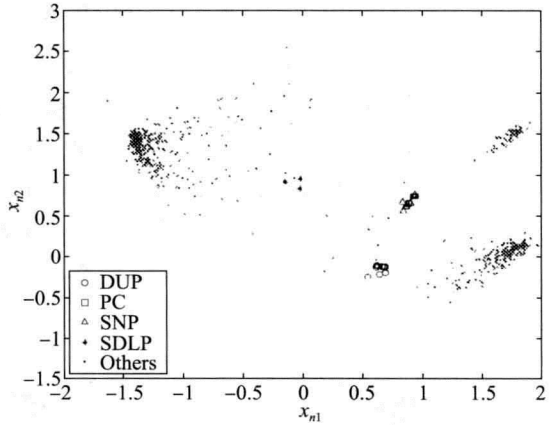


图 7-12 概率二值 PCA 的远离原点的投票数



a) 传统PCA模型中小的政党的可视化



b) VB二值PCA模型下小政党的可视化模型

图 7-13 使用两种 PCA 方法可视化小的政党

最后，由于我们选择模拟缺失值的方法，所以我们对每一个议会成员有一个独立的协方差矩阵 Σ_{x_n} 。从图 7-14 中我们可以看到，用椭圆表示最不确定的 20 个成员的协方差矩阵。这些成员很少投票。很清楚在他们的位置没有真实的模型，这个模型没有把那些没有参与投票的议会成员拉到原点。

可以用这些数据做更有趣的分析，但这超出了本书的范围。重要的一点是，一个基于敏感假设的模型能够正确地处理缺失值（二值概率 PCA），可能比用基础的 PCA 给我们更多议会成员中有趣变量的信息。

7.7.2.1 题外话——与分类的关系

在我们结束本章讨论之前，努力得到关于模型如何工作的直觉是值得的。从表面上看，这个方法似乎不太复杂，但可能是最早把它看做分类模型的方法。训练数据由没有输入特征的 M 个分类标签（对应每一张选票）组成。该模型导出一套隐观测 (x_n) 和 M 个分类函数（由 w_m 定义），以至于我们能够满足尽可能多的类标签。图 7-15 展示了 4 个选票的例子（输入特征）和隐空间的一致决策边界。议会成员用他们的后验均值作为他们在图中的位置，并根据他们选票类型标注（圆或者正方形 ± 1 ，浅灰色的点代表缺失值）。模型已经标注出了议

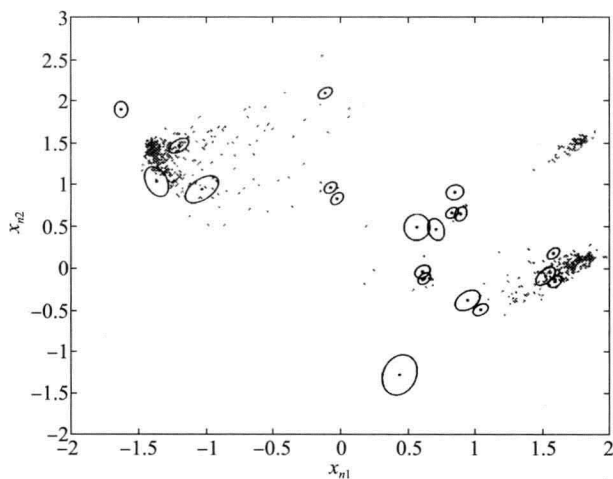


图 7-14 最高不确定性的 20 个议会成员的具体协方差矩阵

会成员在隐变量空间的位置，并以一些分类标签尽可能被满足的方式构造决策边界。满足所有的标签不总是可能的，例如在选票 1 边界右侧的圆。

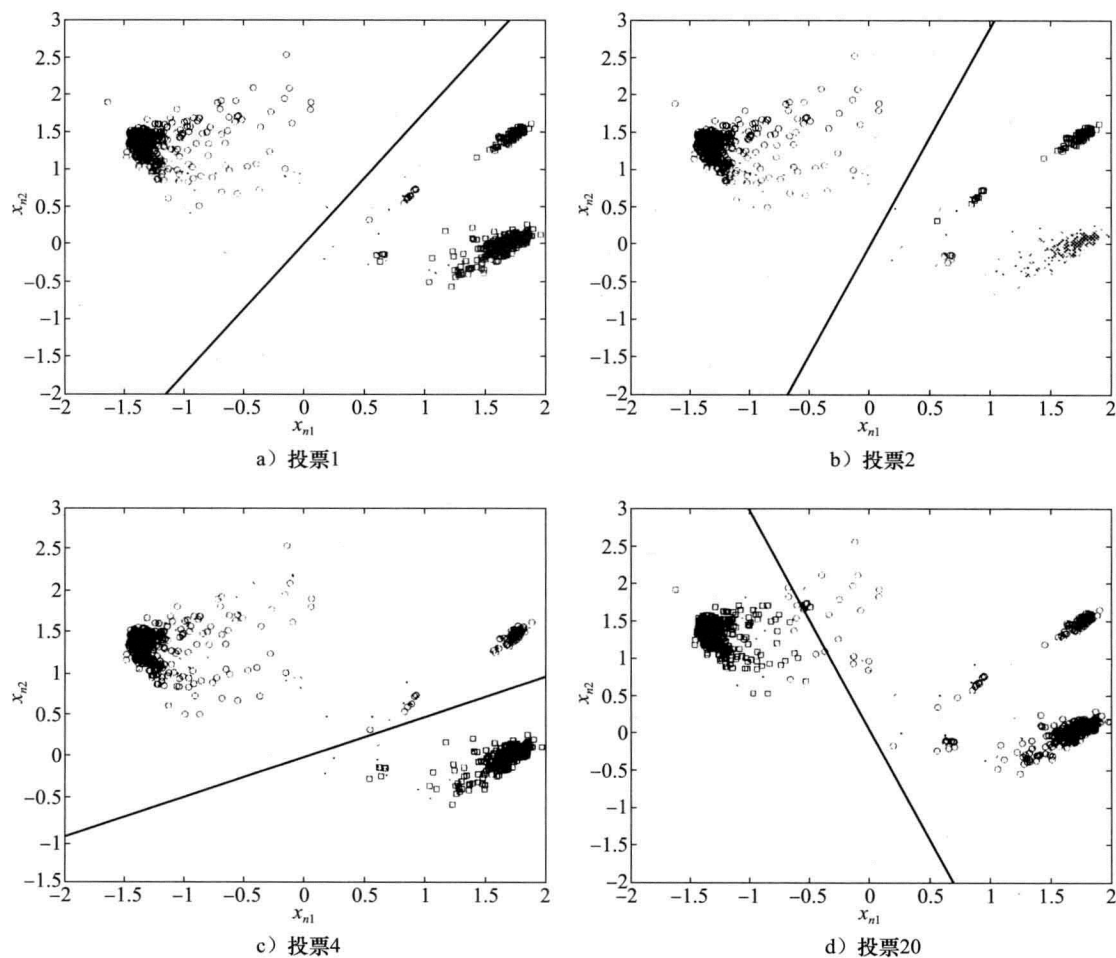


图 7-15 4 种选票的投票情况，每一个议会成员根据他们选票情况被划分为圆或者正方形（浅灰色的点代表没有投票）

7.8 小结

在本章中，我们用主成分分析方法（PCA）和某些概率变量介绍隐变量模型的概念和变分贝叶斯的推理技术。有很多其他的隐变量模型正在用于各种各样的应用中，尤其在信息检索领域应用很多，我们希望再次引入的技术能使读者理解更多具有特别应用的模型。

变分贝叶斯是用于机器学习领域的很流行的推导技术。用某些近似后验技术，我们正在近似精度和计算复杂度之间做一个权衡。根据经验表明，VB 发现了一个在易处理性和精度之间好的权衡。记住其他方法能够用于隐变量模型中执行推断是很重要的——我们已经在第6章看到 EM 算法正在用于混合模型。而且，用辅助变量代替概率似然当然不是我们克服二值似然的唯一方法——正如我们在第4章中看到的。

7.9 练习

EX 7.1 当 $Q(\theta)$ 与真实后验概率 $p(\theta|X)$ 相同时，计算式 (7-6) 中的最大值（即真实的 log 边缘似然）。

EX 7.2 在 $\mathcal{L}(\theta)$ 取下界时，计算 7.5 节给出的概率 PCA 模型中的每一项。

EX 7.3 计算式 (7-15) 给出的带有缺失值的概率 PCA 模型的变分后验概率的每一项。

其他阅读材料

- [1] Christopher M. Bishop. Variational principal components. In *Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99*, pages 509–514, 1999.

使用变分贝叶斯推理的概率主成分模型的一个例子。该文使用先验知识保证隐藏维度的稀疏性，并提供一种避免选择隐空间大小的方法。

- [2] I.T. Jolliffe. *Principal Component Analysis*. Springer, second edition, 2002.

主成分分析方面的综合性参考书。

- [3] Michael Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

- [4] Arto Klami and Samuel Kaski. Probabilistic approaches to detecting dependencies between data sets. *Neurocomputing*, 72:39–46, 2008.

一个有趣的隐变量模型。该模型将概率 PCA 思想扩展到联合分析两个数据集的应用中，同时，提供了 EM 算法和变分贝叶斯的例子。

- [5] S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11(2):305–345, 1999.

一个包含很多实例的关于线性高斯模型的完整可理解的综述。

- [6] Michael Tipping. Probabilistic visualisation of high-dimensional binary data. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, pages 592–598, Cambridge, MA, USA, 1999. MIT Press.

一个概率二值 PCA 算法，算法使用了第4章介绍的对数似然和 EM 推理方法。

- [7] Michael Tipping and Christopher Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(3):611–622, 1999.

一个最大似然的有趣应用。它是最早使用概率方法解决主成分分析的经典统计问题的方法之一。

词汇表

Analytical solution (解析解) 一个数学问题(例如,优化一个量或评价一个积分)的解析解是指某个可以被确切求得的解。我们要处理的多数问题不具有解析解,因此有必要使用迭代算法或采样技术。

Biased (有偏的) 一个估计(例如第2章中的 $\hat{\sigma}^2$)是有偏的,如果它的期望值不等于真实值。

Binomial distribution (二项分布) 一个常用的概率分布,用来描述一个二值实验集合中成功的次数。

Burn-in (老化) 当使用MCMC方法产生样本时,它通常丢掉前面的 N 个点,因为算法可能还没有收敛,所以它们不具有代表性。确定 N 不是轻而易举的。

Conditional independence (条件独立) 在给定条件 C 的情况下,两个(或多个)随机变量 A 和 B 是条件独立的,如果它们的联合分布可以分解为 $P(A, B|C) = P(A|C)P(B|C)$ 。条件独立并不意味着非条件独立。

Conditional probabilities (条件概率) 用来描述依赖其他事件结果的事件概率。例如,如果随机变量 A 的值依赖于随机变量 B 的值,那么给定 B 的值, A 的概率可以写成 $P(A|B)$ 。

Conjugate (共轭) 一个先验与似然是共轭的,如果它们产生的后验具有与先验相同的形式。

Continuous random variables (连续随机变量) 定义在不可枚举样本空间上的随机变量。例如,定义在所有实数上的随机变量。

Convergence (sampler) (收敛 (抽样器)) 如果一个抽样器生成的样本都来自于同一个分布,那么就说这个抽样器收敛了。在抽样器收敛之前生成的样本不应该使用。

Covariance (协方差) 对于多个变量的分布,协方差是方差的一种通用形式。协方差矩阵描述了不同变量之间如何共变的,即它们是如何关联的。

Cross-validation (交叉验证) 一项用于验证和模型选择的技术。将数据随机分为 K 组。然后,对模型训练 K 次,每次留下一组数据。

Decision boundary (决策边界) 在决策问题中分开两类的直线。

Deterministic (确定性的) 非随机的。例如,第1章中的模型, $t = \mathbf{w}^T \mathbf{x}$ 是确定性的。 \mathbf{x} 的相同值总会给出相同的 t 值。

Discrete random variables (离散随机变量) 定义在可枚举样本空间上的随机变量。

Discriminative classifier (判别式分类器) 显式定义(并优化)类别间决策边界的分类器。

Expectation (期望值) 对于一个(离散)随机变量 X , X 的函数 $f(X)$ 的期望值定义为:

$$E_{p(X)}\{f(X)\} = \sum_x P(x)f(x)$$

可以把它看做是根据 X 取不同值可能性的加权平均。对于连续随机变量,将求和变为积分。

Feature selection (特征选择) 在一些分类问题中,减少属性数量是有作用的。这个过程即为特征选择。

Fisher information (Fisher 信息) Fisher 信息用来度量随机变量对某一模型参数提供的信息量大小。

Function (函数) 一种定义两个或多个变量之间关系的方法。例如,

$$t = f(x)$$

告诉我们 t 依赖于 x ——如果知道 x ,就可以计算 t 。

Generalisation (泛化) 泛化是指把从对象集合学习到的结果应用到以前从未见过对象的能力。例如,第1章中的奥林匹克模型,如果它能很好地预测今后的短跑比赛,那么它的泛化能力就好。换句话说,一个具有好的泛化能力的算法应该能对以前未见过的数据进行良好地预测。

Global optimum (全局最优) 一个函数可以有多个极大点(或极小点),全局最优是指其中最大的(或最小的)。

Graphical model (图模型) 概率的图形表示,其中结点对应于随机变量,有向边对应于依赖关系。

Hessian matrix (Hessian 矩阵) 函数对每对变量的二阶偏导矩阵。由19世纪德国数学家Ludwig Otto Hesse提出并用他的名字命名。

Hyper-parameter (超参数) 用来控制在分层贝叶斯模型中另一个参数先验的参数。

Information theory (信息论) 信息的量化研究。特别是, 随机变量的信息含量与其概率分布有关。具有很强不确定性的分布具有高信息含量。

Joint probability (联合概率) 两个随机变量 A 和 B 的联合概率为它们分别取特定值的概率。例如, A 取值 a 和 B 取值 b 的概率, 这个概率写成 $P(A=a, B=b)$ 。

Likelihood (似然) 数据的概率密度 (或离散情况下的分布) 值, 它以模型参数为条件, 这些参数根据该数据进行评估。它是一个数值, 通过调整参数对其优化, 从而获得最大似然解。

Linear (线性) 一个函数 $t=f(x)$ 是线性的, 如果它满足下面的条件:

$$\begin{aligned}f(x_1 + x_2) &= f(x_1) + f(x_2) \\f(ax) &= af(x)\end{aligned}$$

常见的例子是 $f(x)=wx$ 。

Mahalanobis distance (马氏距离) 两个对象 x_n 和 x_m 间的马氏距离定义为:

$$(x_i - x_j)^T A (x_i - x_j)$$

如果用 I 替代 A , 就可以获得标准的欧式距离的平方。矩阵 A 的作用就是对空间的扭曲, 即各个方向上的距离不等。距离点 x_n 具有相同欧式距离的点集构成一个圆。距离点 x_n 具有相同马氏距离点集构成一个椭圆, 其形状由矩阵 A 决定。

Marginal likelihood (边缘似然) 贝叶斯规则的分母。进行模型比较和选择的有用量。

Marginalisation (边缘化) 通过在一个随机变量的所有可能取值空间上对联合分布求和 (连续的情况为积分), 消去联合分布中该随机变量的过程称为边缘化。例如,

$$P(A=a) = \sum_b P(A=a, B=b)$$

Maximum likelihood (最大似然) 一种常用的参数估计方案, 该方法选择使数据似然值最大的参数。

Maximum a posteriori (最大后验) 一种常用的参数值的点估计方法, 通过引入正则先验来扩展最大似然。

Metropolis-Hastings 一种常用的根据密度产生样本的算法, 无需对归一化常数进行评价。

Model complexity (模型复杂度) 用来描述模型复杂程度的术语。例如, $t=w_0+w_1x$ 比 $t=w_0+w_1x+w_2x^2$ 的复杂度低, 因此它无法发现数据中与后者同等复杂的模式。

Model selection (模型选择) 模型选择是指在特定任务中选择哪个模型。备选模型可以来自同一族, 但并不必须是。例如, 如果我们想用多项式函数 $t = \sum_{k=0}^K w_k x^k$, 选择适当的 K 值就是模型选择问题。

Model (模型) 某个过程的数学描述。例如, 在第 1 章中, 我们提出模型 $t=w_0+w_1x$ 来表示 100 米短跑时间与奥林匹克年 x 之间的关系。

Mode (众数) 一个随机变量分布的众数是指最可能的值。

Monotonic function (单调函数) 单调函数是指无限地下降或上升。一个通用例子是 $\log(x)$, 它总是随着 x 的增大而增大。一个有用的性质是使 $f(x)$ 最小的 x 同样使 $\log(f(x))$ 最小。

Monte Carlo approximation (蒙特卡罗近似) 通过从一个近似分布采样来近似一个期望。一个形如

$$E_{p(x)}\{f(x)\} = \int f(x)p(x)dx$$

的期望可以通过下式近似

$$E_{p(x)}\{f(x)\} \approx \frac{1}{S} \sum_{i=1}^S f(x^i)$$

其中, x^1, \dots, x^S 是 S 个来自 $p(x)$ 的样本。

Multinomial distribution (多项分布) 整数向量的常用分布。例如, 如果我们扔 N 次骰子, 并且用一个 6 维向量记录我们得到每一面的次数, 这个向量可以用一个多项分布的随机变量描述。

Natural logarithm (自然对数) 以 e 为底的对数, 这里记作 \log , 但经常记作 \ln 。

Noise (噪声) 认为不是当前问题感兴趣数据中的变异。例如, 由于测量误差引起的随机波动。

Over-fitting (过拟合) 一个模型是过拟合的, 如果它过于复杂, 并且用其过剩的复杂性来拟合噪声。过拟合的模型往往泛化性能很差。

Parameters (参数) 用来定义模型的变量。例如, 下面模型

$$t = w_0 + w_1x$$

具有两个参数—— w_0 和 w_1 。

Partial derivatives (偏导数) 对一个多元函数求偏导即对每一个变量求导数,同时其余变量被看做常数。

例如,如果函数 $t=f(x, y)$ 定义为:

$$t = 2x^2 + 3y^3 + xy$$

那么相对于 x 和 y 的偏导数如下:

$$\frac{\partial f(x, y)}{\partial x} = 4x + y$$

$$\frac{\partial f(x, y)}{\partial y} = 9y^2 + x$$

Plate (盘子) 在图模型中,特定类型随机变量的许多实例的简略表示。

Polynomial (多项式) 一个多项式函数 $t=f(x)$ 具有 $t = \sum_{k=0}^K w_k x^k$ 的形式。常见的例子是 1 阶(线性)多项式

$t = w_0 + w_1 x = \sum_{k=0}^1 w_k x^k$ (称为 1 阶,因为 x 的最高次幂为 1), 2 次(2 阶)多项式 $t = w_0 + w_1 x +$

$w_2 x^2 = \sum_{k=0}^2 w_k x^k$ 。注意 $x^0=1$ 。

Posterior distribution (后验分布) 是指观测到数据后,参数值的分布。

Precision (精度) 在分层贝叶斯模型中,通常用精度比方差更方便。精度可以定义如下:

$$\tau = \frac{1}{\sigma^2}$$

因此,一个具有均值为 μ 、方差为 σ^2 的高斯分布也可以用精度 τ 表示为:

$$\mathcal{N}(\mu, \tau^{-1})$$

Prior distribution (先验分布) 在观测数据之前,根据我们的知识描述参数值的分布。

Probability density function (概率密度函数) 概率密度函数描述了连续随机变量在样本空间上概率质量的分布。概率密度函数必须大于 0,且在整个样本空间上积分为 1。

Probability distribution (概率分布) 用来描述随机变量特性的函数或一组值。

Probability (概率) 一个事件发生的概率是一个 0~1 的值,用来表示该事件发生的可能性。

Projection algorithms (投影算法) 把数据从 M 维空间投影到 D 维空间 ($D \ll M$) 的一系列机器学习算法。

投影技术可以用于可视化 ($D=2$),也可以用于数据预处理,如分类。

Quadratic (二次) 二次函数 $t=f(x)$ 是 x 的最高次数为 2 的多项式函数。例如, $t=x^2$ 和 $t=w_0 + w_1 x + w_2 x^2$ 都是二次函数。

Random events (随机事件) 我们不能(或不想或不需要)给出事件的确定性模型。例如,扔骰子或投硬币。虽然我们不知道这些事件的输出结果,但我们可能知道不同结果的相对似然值。

Random variable (随机变量) 存储随机事件结果的变量。例如,如果我们抛一个硬币,并给变量 X 赋值为 1,若硬币正面朝上;赋值为 0,若背朝上, X 是一个随机变量。

Random walk (随机漫步) 一个样本序列,每个样本都依赖于它前面的那个样本。

Regularisation (正则化) 对参数值加以限制,以限制模型的最大复杂性。

Sample space (样本空间) 一个随机变量所有可能取值的空间。换句话说,是一个特别的随机事件可能输出的集合。

Statistics (统计学) 描述了一系列关于数据收集与解释的方法和原理。

Supervised learning (有监督的学习) 提供了数据对象及相关标记的机器学习任务。

Symmetric matrix (对称矩阵) 一个方阵 \mathbf{X} 是对称的,如果对于所有的 i, j 有 $x_{ij} = x_{ji}$ 。如果是对称矩阵,那么有 $\mathbf{X}^T = \mathbf{X}$ 。

Unbiased (无偏的) 从平均的角度讲,如果一个估计(例如, \hat{w})等于其真实值,则该估计是无偏的。

Unsupervised learning (无监督的学习) 不需要标记数据的机器学习算法。包括聚类和投影。

Validation data (验证数据) 用于帮助选择模型类型和参数的数据,不直接用于训练模型。

Variance (方差) 随机变量和其均值之间差值平方的均值。

索引

A

absolute loss (绝对损失), 5
attributes (属性), 1, 84, 208
auxiliary variables (辅助变量), 265

B

bag-of-words (词袋), 176
Bayes' rule (贝叶斯规则), 49, 98, 120, 140, 170
Bayesian classifier (贝叶斯分类器), 170
Bayesian inference (贝叶斯推理), 139
Bayesian Machine Learning (贝叶斯学习), 98
Bernoulli distribution (伯努利分布), 53, 230
beta distribution (β 分布), 60, 100
bias-variance trade-off (偏差-方差平衡), 75
Binomial distribution (二项分布), 53, 95

C

causality (因果关系), 2
chain rule (differentiation) (链式规则 (微分)), 146
classification (分类), 140, 169
 discriminative versus generative (判别式与产生式), 203
 non-probabilistic (非概率的), 183
 probabilistic (概率的), 170
 text (文本), 175
classification accuracy (分类准确率), 198
clustering (聚类), 207
 similarity measures (相似性度量), 209
combinations (组合), 55
confusion matrix (混淆矩阵), 201
conjugate prior (共轭先验), 102
 non-conjugate models (非共轭模型), 139
covariance (协方差), 52, 78
 Gaussian (高斯), 62
cross-validation (交叉验证), 29, 131, 185, 196, 228
 computational scaling (计算缩放), 32
 leave-one-out (留一法), 31

D

decision boundary (决策边界), 147

definite integrals (定积分), 57
dependence (依赖), 46
Dirichlet distribution (狄利克雷分布), 178

E

Eigenvectors and eigenvalues (特征向量与特征值), 244
evidence (证据), 105
expectation (期望)
 with respect to posterior (相对于后验), 109
expectations (期望), 50
 continuous (连续), 58
 for predictions (面向预测), 98, 152
 with respect to posterior (相对于后验), 129

F

Fisher information (费舍尔信息), 80
function (函数), 2
 linear (线性), 1
 polynomial (多项式), 25
 quadratic (二次), 25

G

Gaussian (高斯), 61
 likelihood (似然), 124
 noise (噪声), 66, 69
 process (过程), 182
 truncated (截断), 267
generalisation (泛化), 28, 34, 74, 75, 196
generative model (产生式模型), 40, 216
graphical models (图模型), 120, 253
 plates (模板), 121

H

hyper-parameters (超参数), 119

I

independence (独立), 46
 in Variational Bayes (在变分贝叶斯中), 251
 multivariate Gaussian (多元高斯), 64

information theory (信息论), 80

J

Jensen's inequality (詹森不等式), 219, 250

K

K-means (K 均值), 208

K-nearest neighbours (K 近邻), 183

kernel density estimation (核密度估计), 163

kernel K-means (核 K 均值), 212

kernel KNN (核 K 近邻), 196

kernel methods (核方法), 186, 193, 212

Kullback-Leibler divergence (Kullback-Leibler 散度), 251

L

Lagrange multipliers (拉格朗日乘法), 188, 223

Laplace approximation (拉普拉斯近似), 149

for logistic regression (面向逻辑回归), 151

latent variables (隐变量), 248

likelihood (似然), 67

binary (二值), 142

classification (分类), 171

in Bayes' rule (在贝叶斯规则中), 99

log (对数), 69

linear (线性)

nonlinear responses (非线性响应), 25

linear model (线性模型), 85

linear modelling (线性建模), 1, 25

logistic regression (逻辑回归), 179

M

margin (间隔), 186

maximisation (最大化), 187

soft (软), 192

marginal distribution (边缘分布), 101

marginal likelihood (边缘似然), 101, 117, 141, 171, 249

matrix (矩阵), 16

determinant (行列式), 64

Fisher information (费舍尔信息), 80

Hessian, 72, 80, 144

identity (单位), 21

inversion (逆), 22

multiplication (乘), 18

notation (符号), 15

symmetric (对称), 71

trace (迹), 88

transpose (转置), 18

maximum likelihood (最大似然), 69

bias of estimator (估计子的偏差), 86, 88

bias of variance estimate (方差估计偏差), 82

maximum-a-posteriori (最大后验), 126, 143, 178, 232

Metropolis-Hastings, 154

minimum loss (最小损失), 6

equivalence to Gaussian ML (等价高斯最大似然), 70

missing data (缺失数据), 260

mixture model (混合模型)

likelihood (似然), 217

mixture models (混合模型), 207, 215

Bayesian treatment (贝叶斯处理), 233

model assumptions (模型假设), 3

model complexity (模型复杂度), 33, 196

model selection (模型选择), 25

difficulty (困难), 31

K-means (K 均值), 210

via marginal likelihood (通过边缘似然), 117

with likelihood (利用似然), 74

with loss (利用损失), 28

Monte-Carlo (蒙特卡罗), 58

mRNA data (mRNA 数据), 208

multinomial distribution (多项分布), 54, 177

multivariate Gaussian (多元高斯), 62

covariance (协方差), 62, 78

independence (独立), 62

N

Naive Bayes (朴素贝叶斯), 175

Naive Bayes classifier (朴素贝叶斯分类器), 175

Newton-Raphson, 144

noise (噪声), 39, 76, 82, 85

additive (可加性), 66

Gaussian (高斯), 85

nonlinear responses (非线性响应), 27

normal (正态), 见 Gaussian

O

over-fitting (过拟合), 28, 33, 34, 74, 75, 196, 228

P

- parameter (参数), 2
- point predictions (点预测), 12, 110
- posterior approximation (后验近似)
 - Laplace (拉普拉斯), 149
 - sampling (采样), 156, 163
- posterior distribution (后验分布), 101
 - exact computation (精确计算), 103, 120
 - expectation with respect to (期望相对于), 109
 - sampling from (从...采样), 127
- predictions (预测), 1
 - uncertainty (不确定性), 84, 85
- Principal Components Analysis (主成分分析), 242
- prior distribution (先验分布), 75, 99
 - choice (选择), 111
 - conjugate (共轭), 139, 173
 - strength (强度), 113, 116
- probability (概率), 39
 - conditional (条件), 44
 - joint (联合), 45
- probit (概率), 265
- projection (投影), 239

R

- random variable (随机变量), 41
 - continuous (连续), 42, 55
 - density (密度), 55
 - discrete (离散), 41
 - distributions (分布), 42
 - marginalisation (边缘化), 47
 - marginalisation, continuous (边缘化, 连续), 58
 - vectors (向量), 52
- regression (回归)
 - logistic (逻辑), 179
- regularisation (正则化), 33, 75
- ROC analysis (ROC 分析), 199
 - AUC (曲线下面积), 200

S

- sampling (采样), 59, 153, 154
 - burn-in (老化), 161
 - convergence (收敛), 161
 - from posterior (从后验), 153
 - visualising output (输出可视化), 163
- sensitivity and specificity (敏感性和特异性), 198
- sigmoid (sigmoid 函数), 142
- smoothing (平滑), 177
- squared loss (平方损失), 4
 - matrix form (矩阵形式), 19
 - minimising (最小化), 6
- Support Vector Machines (支持向量机), 186

T

- Taylor expansion (泰勒展开), 150
- turning points (拐点), 6

U

- uncertainty (不确定性), 48
 - in parameters (参数中), 39, 76, 78, 80, 82, 148
 - in predictions (预测中), 39, 83, 85, 152
- uniform distribution (均匀分布), 58

V

- validation (验证), 29
- variance (方差), 51
 - reduction in posterior (在后验中减少), 105
- Variational Bayes (变分贝叶斯), 249
- vector (向量), 16
 - differentiation with respect to (关于...的微分), 20
 - indexing (索引), 17
 - inner product (内积), 18
 - transpose (转置), 16

推荐阅读



中文版
第6版

作者: Abraham Silberschatz 著
中文翻译版: 978-7-111-37529-6, 99.00元
英文精编版: 978-7-111-40086-8, 69.00元
本科教学版: 978-7-111-40085-1, 59.00元



中文版
第3版

作者: Jiawei Han 等著
英文版: 978-7-111-37431-2, 118.00元
中文版: 978-7-111-39140-1, 79.00元



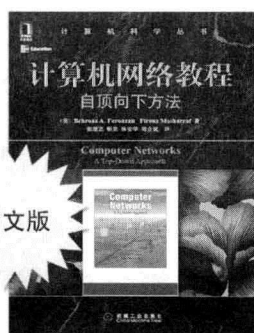
英文版
第3版

作者: Ian H. Witten 等著
英文版: 978-7-111-37417-6, 108.00元
中文版预计2014年3月



英文版
第5版

作者: Andrew S. Tanenbaum 著
书号: 978-7-111-35925-8, 99.00元



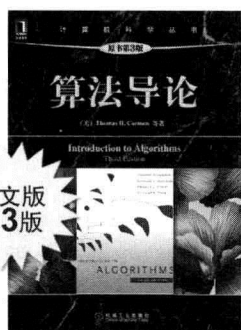
中文版

作者: Behrouz A. Forouzan 著
英文版: 978-7-111-37430-5, 79.00元
中文版: 978-7-111-40088-2, 99.00元



中文版
第4版

作者: James F. Kurose 著
书号: 978-7-111-16505-7, 66.00元



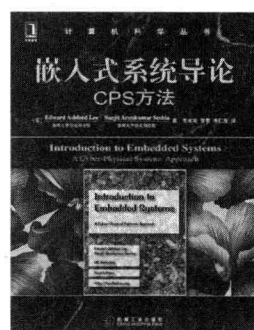
中文版
第3版

作者: Thomas H. Cormen 等著
书号: 978-7-111-40701-0, 128.00元



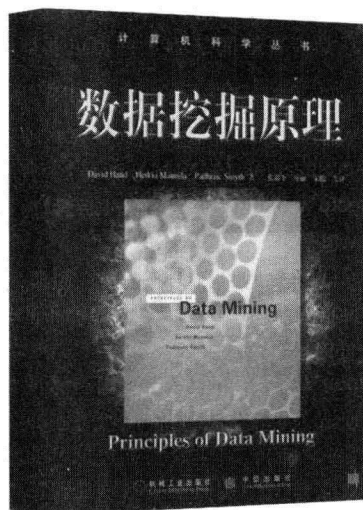
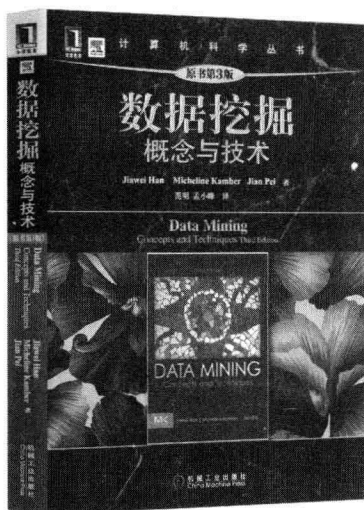
英文版
第5版

作者: John L. Hennessy 著
书号: 978-7-111-36458-0, 138.00元



作者: Edward Ashford Lee 著
书号: 978-7-111-36021-6, 55.00元

推荐阅读



数据挖掘：概念与技术（第3版）

作者：Jiawei Han 等 译者：范明 等 ISBN: 978-7-111-39140-1 定价：79.00元
英文版：978-7-111-37431-2 定价：118.00元

数据挖掘原理

作者：David Hand 等 译者：张银奎 等 ISBN: 978-7-111-11577-5 定价：48.00元

数据挖掘导论（英文版）

作者：Pang-Ning Tan 等 ISBN: 978-7-111-31670-1 定价：59.00元

社交网站的数据挖掘与分析

作者：Matthew A. Russell 译者：师善 ISBN: 978-7-111-36960-8 定价：59.00元

机器学习

作者：Tom Mitchell 译者：曾华军等 ISBN: 978-7-111-10993-7 定价：35.00元
英文版：7-111-11502-3 定价：58.00元

数据挖掘：实用机器学习工具与技术（英文版·第3版）

作者：Ian H. Witten 等 ISBN: 978-7-111-37417-6 定价：108.00元
中文版：预计2013年12月出版